Nr. 7/86 Juli
DM 6.50, sfr 6.50, öS 50, Lit 5900, hfl 7.50

peeker

Registerprogramm
Memory-Spiel
ProDOS-Bibliothek
Mousewrite
Laser 128
RGB-Farbkarte

Hüthig
PUBLIKATION

Leser werben Leser

»peeker« bietet Ihnen etwas Neues!
Wer jetzt schenkt hat mehr von seinem Apple.

Dafür schenkt »peeker« Ihnen etwas: Die praktische und formschöne Box, die Ihre Disketten übersichtlich ordnet.
Sie wissen ja, wie gut »peeker« Ihnen im täglichen Umgang mit Ihrem Apple behilflich ist. Und Sie brauchen Ihren »peeker« nicht mehr zu teilen oder auszuleihen.
Die Diskettenbox ist unser Geschenk an Sie für einen neuen »peeker«-Abonnenten. Denn wer einen Apple hat, der soll auch seinen »peeker« haben.

peeker 7/86

Bestellcoupon

Ich habe den neuen Abonnenten geworben und erhalte kostenlos die Diskettenbox.

Name, Vorname

Straße, Postfach

PLZ, Ort

X
Datum, Unterschrift

Ich bin der neue Abonnent. Bitte liefern Sie mir bis auf Widerruf, zumindest aber für 1 Jahr, »peeker« zum Jahresbezugspreis von z. Zt. DM 72,– (Ausland plus DM 18,– Porto) an folgende Anschrift:

Name, Vorname

Straße, Postfach

PLZ, Ort

X
Datum, 1. Unterschrift

Gewünschte Zahlungsweise
☐ gegen Rechnung
☐ bargeldlos durch Bankeinzug

Konto-Nr. Bankleitzahl

Geldinstitut

Vertrauensgarantie:
Diese Bestellung kann ich innerhalb einer Woche bei Dr. Alfred Hüthig Verlag GmbH, Im Weiher 10, 6900 Heidelberg 1 widerrufen. Zur Wahrung der Frist genügt die rechtzeitige Absendung. Ich bestätige die Kenntnisnahme mit meiner Unterschrift:

X
2. Unterschrift

Coupon ausschneiden und einsenden an:

»peeker«
Abonnementservice
Im Weiher 10
6900 Heidelberg 1

editorial

Peeker-Umfrage

Wie Sie wissen, wird im Herbst dieses Jahres das Apple-II-Nachfolge-Modell erscheinen. Wir haben Apple-München zugesichert, daß wir vor der offiziellen Vorstellung keine technischen Details durchsickern lassen, doch können Sie sich natürlich denken, daß wir bereits mehrere Beiträge zum zukünftigen Apple in Vorbereitung haben.

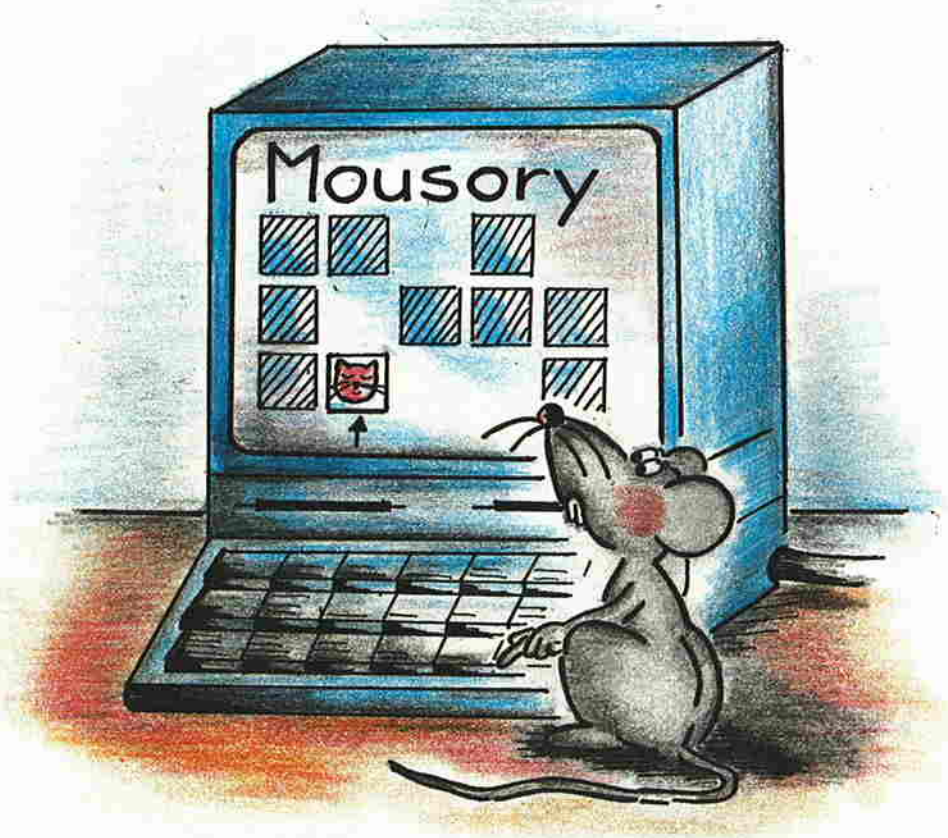
In diesem und den beiden nachfolgenden Peeker-Heften befindet sich jeweils auf Seite 45 eine Umfragekarte. Wir hoffen, daß sich möglichst viele Leser an dieser Umfrage beteiligen werden, denn dadurch können Sie selbst Einfluß auf die redaktionelle Ausrichtung des Peekers nehmen. Beispielsweise bekundeten Sie in der letztjährigen Peeker-Umfrage starkes Interesse an Pascal, dem wir durch entsprechende Kurse und Aufsätze Rechnung getragen haben.

Die Apple-Landschaft hat sich beachtlich verändert. Deshalb ist der wichtigste Punkt in der diesjährigen Umfrage Ihr Interesse an den zukünftigen Apple-Computern (Apple-II-Nachfolger, Mac-Nachfolger) und den z. Zt. wichtigsten Nicht-Apple-Computern (IBM-PC, Atari-ST). Denn obwohl die Zahl der Apple-II-Besitzer nach wie vor sehr groß ist, sind Abwanderungen in Richtung IBM-PC und Atari-ST unverkennbar:

– Der IBM-PC in seinen verschiedenen Varianten (XT, AT, Kompatible usw.) hat sich inzwischen im kommerziellen und technischen Bereich als Arbeitspferd etabliert. Der IBM-PC strahlt keine innovative Technik, sondern Solidität und Beständigkeit aus. Aus genau diesem Grunde hat der Macintosh gegenüber dem IBM-PC keine Chance gehabt. So wurden etwa allein vom Siemens-PC im letzten Jahr mehr Geräte als vom Macintosh verkauft.
– Der Atari-ST konnte sich inzwischen als anspruchsvolles Freak-Gerät durchsetzen, doch blieb ihm bislang der kommerziell-technische Einsatz versagt. Dies könnte sich jedoch in Zukunft ändern.
– Der Macintosh wird immer seine Abnehmer finden, aber seine Verbreitung war und bleibt auf Sondermärkte (Manager-Tool, In-House-Publishing usw.) beschränkt.
– Der alte Apple II war und ist im technischen und Small-Business-Sektor sowie im schulischen und Hobby-Bereich im Einsatz. Nun wird es darauf ankommen, ob der Apple-II-Nachfolger nicht nur von den Alt-Besitzern, also dem Gros der Peeker-Leser, akzeptiert wird, sondern sich auch neue Märkte erschließen kann.

Ulrich Stiehl

inhalt



Impressum

Peeker
3. Jahrgang 1986
ISSN 0176–9200
© für den gesamten Inhalt einschließlich der Programme Dr. Alfred Hüthig Verlag, Heidelberg 1986
Verleger und Herausgeber: Dipl.-Kfm. Holger Hüthig
Geschäftsführung Zeitschriften: Heinz Melcher
Chefredakteur: Ulrich Stiehl (us)
Redaktion: Dagmar Berberich

Telefonnummern:
Zentrale: 06221/489-1
Redaktion: 06221/489-352
Anzeigen: 06221/489-206
Abonnement: 06221/489-283
Software: 06221/489-231
Bücher: 06221/489-353
(Bestellungen bitte nur schriftlich)

Abonnement:
Der Abonnent kann seine Bestellung innerhalb von 7 Tagen schriftlich durch Mitteilung an den Dr. Alfred Hüthig Verlag GmbH, Postfach 102869, 6900 Heidelberg 1, widerrufen. Zur Fristwahrung genügt die rechtzeitige Absendung des Widerrufs (Datum des Poststempels). Das Abonnement verlängert sich zu den jeweils gültigen Bedingungen um ein Jahr, wenn es nicht zwei Monate vor Jahresende schriftlich gekündigt wird. Die Abonnementsgelder werden jährlich im voraus in Rechnung gestellt, wobei bei Teilnahme am Lastschriftabbuchungsverfahren über die Postscheckämter und Bankinstitute eine vierteljährliche Abbuchung möglich ist.
Nichterscheinen infolge höherer Gewalt berechtigt nicht zu Ansprüchen gegen den Verlag.

Hüthig PUBLIKATION

PEEKER

Heft 7/1986

Praxis

Registerprogramm
für Buchregister und zweisprachige Glossare
von Ulrich Stiehl 6

Hobby

Mousory
Memory-Spiel mit der Apple-Maus
von Wolfgang Landgraf 12

Applesoft

Verkettung von Applesoft-Programmen
Variablenübergabe unter DOS 3.3
von Frank Bühler 18

Hardware

ProDOS-Uhrkartentreiber
Realtime-Clock "Time II"
von Dipl.-Ing. Achim Sucker 22

ProDOS

EDIT
Ein Luxus-Editor for ProDOS
Teil 2: Dumpen und Editieren
von Arne Schäpers 24

ProDOS-Runtime-Library
Eine Muster-Bibliothek für Assembler-, Applesoft- und Pascal-Programme
von Ulrich Stiehl 40

Macintosh

Datenübertragungsraten beim Machintosh
von Arne Schirmacher 54

Praxis

Schreiben mit der Maus
Textverarbeitungsprogramm Mousewrite
getestet von Thomas Bühner 56

Anschluß eines Brother HR-15 XL II
an den Apple IIc
von Hans-Peter Sprauer 59

Bücher 60

Produkte

Laser 128
Ein Apple-IIc-Kompatibler
getestet von Hans-Martin Eng 64

Keycom 6
Eine externe Tastatur für den Apple IIe
getestet von Hans-Martin Eng 65

RGB-Farbinterfacekarte AI-80Z
Ein Erfahrungsbericht
von Dirk Katzschke 67

Star Gemini-15
getestet von Jürgen Geiß 70

Anschrift:
Dr. Alfred Hüthig Verlag GmbH
Im Weiher 10, Postfach 10 28 69
6900 Heidelberg
Telefon (0 62 21) 4 89-1
Telex 4-6 17 27 hued d.
Telefax (0 62 21) 489 279
BTX * 51851 #

Auslieferung für die Schweiz:
Delta-Verlag
Herr R. de Forest
Gugelmattstraße 31
8967 Widen
Telefon 057 / 33 86 86

Vertrieb:
Erscheinungsweise: 12 Hefte jährlich,
Erscheinungstag jeweils 1 Woche vor Monatsbeginn.
Jahresabonnement Inland DM 72,–, einschl. MwSt und Versandkosten.
Jahresabonnement Ausland DM 72,– plus DM 18,– Versandkosten.
Einzelheft DM 6,50
Vertrieb Handel:
MZV – Moderner Zeitschriften Vertrieb GmbH
Breslauer Str. 5, Postfach 1123,
8057 Eching b. München,
Tel. 089 / 31 90 06 13, Telex 0 522 656
Vertriebsleitung:
Walter Menzel, Tel. (0 62 21) 48 92 80

Bankverbindungen:
Zahlungen: an den Dr. Alfred Hüthig Verlag GmbH, D-6900 Heidelberg 1: Postgirokonten: BRD: Ludwigshafen 4799-673, BLZ 545 100 67; Österreich: Wien 75558 88; Schweiz: Basel 40-24417; Niederlande: Den Haag 1 457 28; Italien: Mailand 5 968 92 08; Belgien: Brüssel 1084 1261; Dänemark: Kopenhagen 603 4969; Norwegen: Oslo 199 4243; Schweden: Stockholm 5477 76-5
Bankkonten: Landeszentralbank Heidelberg 67 207 341; BLZ 672 000 00; Deutsche Bank Heidelberg 02 65 041; BLZ 672 700 03; Bezirkssparkasse Heidelberg 204 51, BLZ 672 500 20.

Herstellung:
Produktionsleitung: Gunter Sokollek
Gestaltung: Rainer Schmitt
Titelbild: Werner Hable
Satz und Druck:
Heidelberger Verlagsanstalt
Printed in Germany

„Peeker" ist eine unabhängige Zeitschrift. Sie ist nicht verbunden mit der Firma Apple Computer, Inc. oder der Apple Computer GmbH. APPLE, das Apple-Zeichen und MAC sind Warenzeichen der Firma Apple Computer, Inc. und MACINTOSH ist ein Warenzeichen, in Lizenz vergeben von der Firma McIntosh Laboratory an die Firma Apple Computer, Inc.

Registerprogramm

für Buchregister und zweisprachige Glossare

von Ulrich Stiehl

Die Urversion dieses Programms, die zunächst nur aus TASC-compilierten Applesoft-Modulen bestand, ist 1982 auf einer Club-Diskette der AUGE erschienen. Später wurde das Programmpaket durch Assemblerroutinen für alle zeitkritischen Programmteile sowie durch einen komfortablen Zeileneditor für den Registerredigierer erweitert und in verschiedenen Varianten bei Verlagen und Druckereien besonders für Buchregister (= Stichwortregister mit Seitenzahlen) praktisch eingesetzt. Bei der hier vorgestellten und auf der Peeker-Sammeldisk #19 befindlichen Programmversion ist der Gesamtumfang auf 50% *einer* externen DOS-3.3-Datendiskette beschränkt. Das bedeutet praktisch ca. 65K Gesamtumfang bei Disk-II-140K-Disketten (gilt auch für MDB-Festplatten und 256K-RAM-Karten) sowie ca. 100K Gesamtumfang bei Erphi-640K-Disketten. Für Buchregister und kleinere Glossare ist dies ausreichend.
Dagegen muß man bei großen Wörterbüchern von vornherein mehrere Registerdisketten (z.B. A-G, K-L, M-Z o.ä.) anlegen. Auch hierfür wurden Module entwikkelt (Registervorsortierer usw.), doch ist das Registerprogramm dann nur noch für Disk-Jockeys geeignet. Die hier beschriebene Programm-Variante ist somit für den „Hausgebrauch" am komfortabelsten.

1. Theorie

Der Zweck des Registerprogramms besteht im Erstellen, Redigieren, Sortieren, Mischen, Ausdrucken und Konvertieren von Buchregistern, zweisprachigen Glossaren und sonstigen Registern, die sich aus Doppelstrings (Sortierwort + Information) zusammensetzen. Das Sortierwort nennen wir auch kurz „Wort" („W" im Registerredigierer), und die Information nennen wir „Info" („I" im Registerredigierer).

Beispiel Buchregister
Schreibsatz 20, 25, 37
Schriftart 39
Schriftenreihe 245, 249
Schriftfamilie 41, 47 usw.
Hier sind die Sortierwörter die Stichwörter, und die Infos sind die Seitenzahlen.

Beispiel Wörterbuch
vollenden complete, finish
vollständig complete, entire
Volumen bulk
voluminös bulky
Vorabdruck first serialization usw.
Hier sind die Sortierwörter die deutschen Wörter, und die Infos sind die englischen Wörter. Zweisprachige Glossare können übrigens vollautomatisch umgedreht werden (deutsch-englisch → englisch-deutsch).

1.1. Wort und Info

Abstrakt formuliert besteht ein Register im Sinne dieses Programms aus einer Folge von Doppelstrings = Wort + Info, Wort + Info usw.:

- Sortierwörter und Infos können je maximal 77 Zeichen lang sein, z.B. „Mehrfarbenhochleistungsrollenrotationsmaschine" = 46 Zeichen. Doch sollte eine *durchschnittliche* Länge von ca. 40-50 Zeichen für Wörter und Infos nicht überschritten werden, da der Stringpool nur etwa 25K-30K umfaßt. Wenn die Info eine Seitenzahl ist (= Buchregister), so wird die Feldlänge auf 4 Stellen begrenzt (0-9999). Der Registerredigierer läßt dann im Info-Feld nur Zahlzeichen zu, so daß man nicht etwa statt Nullen O's eingeben kann.
- Wörter sollten aus Buchstaben bestehen. Sonderzeichen (z.B „Goethe's Werke"; Sonderzeichen hier: Apostroph und Leertaste) sind erlaubt, da sie beim Sortieren ignoriert werden (Sortierwert null). Ziffernhaltige Wörter (z.B. „HN3", „H2O" usw.) sind zulässig, doch erfolgt die Sortierung von Zahlzeichen im String nach dem ASCII-Code. Notfalls muß man Wörter, die reine Zahlen darstellen, auf gleiche Stellenanzahl bringen, indem man links mit Nullen oder Leertasten auffüllt (z.B. „0010" oder „□□10").
- Die Info kann wahlweise aus Buchstaben oder Zahlen bestehen. Je nach Option erfolgt dann bei Buchstaben eine alphabetische und bei Zahlen eine numerische Sortierung in aufsteigender Folge (A,B,C... bzw. 1,2,3...).
- Kontroll-Zeichen (z.B. Crtl-D usw.) sind weder beim Sortierwort noch bei der Info zulässig.
- Aus Gründen der Kompatibilität mit Applesoft können keine Gänsefüßchen eingegeben werden. Dafür sind Doppelpunkte und Kommas erlaubt. Anstelle des Gänsefüßchens nehme man den Apostroph.

- Weder Sortierwort noch Info sollten ausschließlich aus Sonderzeichen bestehen (z.B. „----" = Sortierwert null), doch kann man die (Buchstaben-)Info ersatzweise mit z.B. „-" eingeben, wenn die richtige Info erst später eingesetzt werden soll. Man beachte nämlich, daß Leerstrings vom Registerredigierer (zwecks Vermeidung von „Registerleichen") nicht akzeptiert werden, d.h. Wort und Info müssen aus mindestens 1 Zeichen bestehen.

1.2. Sortierreihenfolge

Das Registerprogramm berücksichtigt folgende Zeichensätze:
- reine Großschreibung beim alten Apple II; dann können allerdings auch nur Großbuchstaben eingegeben werden;
- amerikanischen ASCII-Zeichensatz mit Groß- und Kleinschreibung und eckigen sowie geschweiften Klammern (= Akkoladen);
- deutschen Zeichensatz mit Groß- und Kleinschreibung, Umlauten und ß.

Sortierung der Sortierwörter

Der besondere Vorzug dieses Programms besteht darin, daß die Sortierwörter nach Codiertabelle sortiert werden. Welchen Vorzug dies hat, wird Ihnen erst dann bewußt, wenn Sie eines der üblichen Datenbankprogramme (z.B. dBase) für Buchregister einsetzen wollen. Die interne Konvertierung nach der Codiertabelle verändert selbstverständlich *nicht* die eingegebenen Strings; vielmehr treten die intern umcodierten Sortierwörter für den Programmbenutzer niemals in Erscheinung. (Sie werden auch nicht auf Diskette zwischengespeichert, sondern nur im RAM erzeugt.)
Für den **deutschen** Zeichensatz wird die folgende Codiertabelle benutzt:

```
          abcdefghijkl mnopqrstuvwxyz
          ä            ö     ß ü
0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ
          Ä            Ö     Ü
-------------------------------------
0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

- Großbuchstaben und Kleinbuchstaben haben denselben Sortierwert (z.B „Änderung“ steht nach „ändern“).
- Umlaute werden gemäß Duden wie Nichtumlaute behandelt (z.B. wird „Hündin“ wie „HUNDIN“ sortiert).
- ß wird in S umgewandelt. Bei früheren Versionen wurde ß in SS konvertiert, doch sind die diesbezüglichen Sortierfolgen derart selten (z.B. „Ha/se/nfilet“, „ha/sse/n“, „ha/ße/rfüllt“), daß man die unkorrekte Sortierung in Kauf nehmen kann (z.B. hier nach interner Umcodierung: 1. HASENFILET, 2. HASERFULLT, 3. HASSEN).
- Sonderzeichen werden beim Sortieren ignoriert („Goethe's Werke“ wird wie „GOETHESWERKE“ sortiert). Sonderzeichen sind
!#$%&/()=?'+*# ↑ ,;.:-<>
sowie die Leertaste zwischen Wörtern.
- Zahlen *innerhalb* eines im übrigen aus Buchstaben bestehenden Sortierwortes haben einen niedrigeren Sortierwert als A und werden nur dann richtig sortiert, wenn sie am Anfang oder Ende des Sortierwortes stehen und *dieselbe* Ziffernanzahl haben, z.B. „1.Haus“ kommt vor „2.Hausbesitzer“. Aber „10.Haus“ käme nach „9.Hausbesitzer“, da „1“ < „9“ (als erstes ASCII-Zeichen der Strings). Ziffernhaltige Wörter sind deshalb möglichst zu vermeiden, und bei numerierten Sortierwörtern ist Vorsicht geboten.

Die **amerikanische** Codiertabelle unterscheidet sich von der deutschen lediglich darin, daß die anstelle der Umlaute und des ß benutzten Sonderzeichen (eckige und geschweifte Klammern usw.) ebenfalls keinen Sortierwert haben.
Der alte Apple-II-Großbuchstaben-Zeichensatz wird nach der amerikanischen Codiertabelle behandelt, wobei Kleinbuchstaben ohnehin nicht vorkommen.

Sortierung der Infos

Die Infos werden je nach Option entweder als Zahlen (= Seitenregister: von 0-9999 numerisch aufsteigend) oder wie die Sortierwörter nach Codiertabelle (= sonstige Register) untersortiert. Untersortierung bedeutet, daß beim Ausdruck die Infos zu den gleichen Sortierwörtern sortiert zusammengefaßt werden:

	WORT	INFO
aus	Haus	70
	Haus	20
	Haus	40
wird	Haus	20, 40, 70

Die Kommas werden automatisch eingefügt.

1.3. Dateistruktur

Eine Registerdatei hat auf der Diskette stets folgende Struktur (Beispiel):

000150	Doppelstringanzahl
"Anton	1. Wort
"20	1. Info
"Berta	2. Wort
"72	2. Info
"...	(3.-186. Wort)
"...	und Info)
"150. Wort	letztes Wort
"150. Info	letzte Info
ULI	Wort-Endmarker
ULI	Info-Endmarker

Jede Registerdatei wird mit einer 6stelligen Zahl (= Anzahl der Doppelstrings) eingeleitet und mit dem Doppel-Endmarker beendet. Sortierwörter und Infos werden mit " eingeleitet, das beim Ausdruck usw. jedoch nicht in Erscheinung tritt. Neben den Wörtern werden auch die Zahlen stets als Strings abgespeichert. Registerdateien können von *erfahrenen* Programmierern auch mit jedem beliebigen Textprogramm (z.B. Applewriter, Fastwriter usw.) erstellt und/oder redigiert werden, falls man das charakterisierte Textfile-Format exakt beachtet und unzulässige Zeichen vermeidet. Der Registerredigierer bietet jedoch allen erforderlichen Redigierkomfort.

Wort und mehrere Infos

Um das Umdrehen bei Wörterbüchern sowie das Untersortieren bei Registern schlechthin zu gewährleisten, müssen die *gleichen* Sortierwörter sooft eingegeben werden, wie zugehörige Infos vorkommen. Ist die Untersortierung jedoch bereits bekannt oder unerheblich, dann kann man ausnahmsweise mehrere Infos *einem* Sortierwort – durch Kommas getrennt – zuordnen, z.B.
Stiel 72, 83, 105
oder z.B.
Stiel handle, stem, stick
Bei Buchregistern werden normalerweise *mehrere* Seitenzahlen zu dem *gleichen* Stichwort in der Form (Beispiele)
Wort 10, 15-16, 100-105
(= Streckenstrich-Version) oder
Wort 10, 15p., 100pp.
(= et-cetera-perge-Version)
zusammengefaßt. Da die auf der Sammeldisk #19 enthaltene Registerprogrammversion für Buch- *und* Fremdsprachenregister gedacht ist, werden die Infos nicht zusammengefaßt. (Bei Bedarf kann die Streckenstrich-Version gegen Vergütung gesondert bezogen werden.)
Die Anzahl der Unterstichwörter bzw. Seitenzahlen zu dem gleichen Sortierwort ist auf 100 begrenzt.

2. Praxis

2.1. Gerätekonfiguration

Das Registerprogramm kann auf einem IIe oder IIc sowie auf einem II+ mit Language-Card eingesetzt werden. Es wird nur die 40-Z/Z-Darstellung benutzt, so daß eine Videx-Karte vor dem Programm ausgeschaltet werden muß. Besitzer eines II+ ohne Kleinschreib-Umrüstsatz können das Programm zwar benutzen, doch können dann natürlich keine Kleinbuchstaben eingegeben werden.
Zwei Disk-Drives in *demselben* Slot sind zwingend erforderlich. *Vor* dem Programmstart muß sich die Programmdiskette in Drive 1 und die Datendiskette in Drive 2 befinden. Während des Programmablaufs ist dann kein Diskettenwechsel mehr erforderlich. Neben den normalen 140K-Disk-II-Laufwerken können auch Erphi-160-Spur-Laufwerke sowie die Megaboard-Festplatte MDB10 bzw. MDB20 Verwendung finden. Ferner können 256K-RAM-Karten benutzt werden.
Für den Registerdrucker genügt ein beliebiger Drucker in einem beliebigen Slot, doch kann man auch testweise auf den Bildschirm „drucken“ (Slot 0; beim IIe/c auch Slot 3). Als Steuerzeichen wird nur Ctrl-L für Blattvorschub verwendet.

2.2. Diversi-DOS 2C

Das Registerprogramm setzt sich aus folgenden Teilprogrammen zusammen:
A 002 STIEHL (Erst-Helloprogramm)
A 003 HELLO (Zweit-Helloprogramm)
B 006 DDMOVER (Diversi-DOS 2C)
B 050 REGISTER.UTILITIES
B 024 REGISTER.RUNTIME
B 009 REGISTERSTARTER.OBJ
B 030 REGISTERREDIGIERER.OBJ
B 018 REGISTERSORTIERER.OBJ
B 019 REGISTERMISCHER.OBJ
B 020 REGISTERDRUCKER.OBJ
B 018 REGISTERUMWANDLER.OBJ
A 007 REGISTERTESTER
A 008 REGISTERUMDREHER
A 007 REGISTERTEILER

Die vorliegende Variante des Registerprogramms ist für **Diversi-DOS 2C** bestimmt (zu anderen Konfigurationen siehe unten), das auf der Sammeldisk #6 (vgl. auch Peeker, 10/85, S. 69) enthalten ist. Booten Sie also zunächst eine mit 2C formatierte Diskette, laden Sie mit LOAD STIEHL das Erst-Hello-Programm von der Sammeldisk #19, initialisieren Sie dann eine Leerdiskette mit INIT STIEHL, und schließlich kopieren Sie mit FID alle anderen der o.g. Programme auf die Leerdiskette, die somit zu Ihrer bootfähigen *Registerprogrammdiskette* wird.
Danach formatieren Sie eine weitere Leerdiskette, und zwar im Falle von 140K-Disketten am besten mit dem Programm DOSLOS aus „Apple DOS 3.3 – Tips und Tricks", S. 167, damit Sie 8192 Bytes mehr gewinnen. Diese Leerdiskette wird somit zur Ihrer *Registerdatendiskette*.
Warnung: Verwenden Sie in keinem Fall Diversi-DOS 4C, weil das in die LC geschobene 4C von einem Teil der Register.Utilities überschrieben werden würde!

2.3. Programmstart

Booten Sie nunmehr Ihre konfigurierte Registerprogrammdiskette, die dann automatisch das Erst-Helloprogramm STIEHL lädt, das u.a. prüft, ob 48K-Diversi-DOS 2C vorliegt. Wenn ja, wird automatisch der DOS-Mover DDMOVER aufgerufen, der Diversi-DOS in die Bank 2 der Language-Card schiebt und danach automatisch das Zweit-Helloprogramm HELLO einlädt. Wenn Sie jetzt oder später eines der Module über „0 = Ende" verlassen, so können Sie das Programm mit
RUN HELLO, D1
stets neu starten. Ein Kaltstart ist dann nicht mehr erforderlich. Das Hello-Programm hat folgende Optionen:
1 Hauptprogramm
5 Registertester
6 Registerumdreher
7 Registerteiler
0 Ende

Über „1" gelangen Sie in ein Zwischenmenü, das nach
(a) deutschem bzw.
(b) amerikanischem
Zeichensatz fragt, und dann gelangen Sie endlich in das Hauptmenü, von dem aus Sie dann wählen können:
1 Registerredigierer
2 Registersortierer
3 Registermischer
4 Registerdrucker
5 Registerumwandler
0 Ende

Sonderkonfigurationen

HELLO lädt bei Option „1" erst die Register.Utilities, die teilweise in die Language-Card, Bank 1, verschoben werden, und dann Register.Runtime, das u.a. die TASC-Runtime enthält, nach deutsch/amerikanisch fragt und dann Registerstart nachlädt, welches das Verbindungsprogramm für Registerredigierer, Registersortierer, Registermischer, Registerdrucker und Registerumwandler darstellt.
Die anderen Module Registertester, Registerumdreher und Registerteiler werden direkt von HELLO gestartet und führen direkt zu HELLO zurück. Diese zeitunkritischen Module sind übrigens nicht compiliert worden, damit Sie auf Programm-Muster für die Entwicklung von eigenen Modulen zurückgreifen können.

MDB-Festplatte: Wenn Sie eine Megacore- oder MDB-Festplatte besitzen, so kopieren Sie Ihre in obiger Weise erstellte Registerprogrammdiskette auf S7, Vv, D1 und Ihre Registerdatendiskette auf S7, Vv, D2 (v = beliebige Volume-Nummer). Im übrigen können Sie dann genau wie oben verfahren, d.h. mit PR#7 von dem entsprechenden Volume booten, denn bei der MDB funktioniert in die LC gemovtes Diversi-DOS 2C ohne Einschränkung.

Erphi-Drives: Bei Erphi-640K-Doppellaufwerken müßten Sie analog verfahren können. Dies wurde jedoch von mir nicht ausprobiert, da ich die für die Erstellung der Backup-Utilities (s. Peeker 6/86) ausgeliehenen Erphi-Drives bereits zurückgeschickt habe.

RAM-Karte: Bei einer 256K-RAM-Disk, die mit Drive 1 und 2 angesprochen werden kann, wird es komplizierter, denn möglicherweise existiert kein RAM-Disk-Driver für in die LC gemovtes Diversi-DOS 2C. Falls dies trotzdem der Fall sein sollte, dann verfahren Sie nach der Anleitung zur RAM-Karte und starten Sie anschließend das Registerprogramm mit RUN HELLO, nachdem Sie alle Files in das Drive 1 der RAM-Disk kopiert haben.

48K-DOS: Wenn Sie zwar eine Language-Card besitzen, aber kein Diversi-DOS 2C, dann können Sie auch mit normalem 48K-DOS-3.3 arbeiten. Das Registerprogramm wird dann mit RUN HELLO und nicht mit RUN STIEHL gestartet. Bei 48K-DOS müssen Sie sich jedoch auf einen Memory-Overflow gefaßt machen, wenn Sie Register eingeben und sortieren, bei denen Sortierwort und Info eine durchschnittliche Gesamtlänge von jeweils mehr als 30 Zeichen haben.

2.4. Registerredigierer

Der Registerredigierer dient zum (1) Neueingeben, (2) Ändern, (3) Weitereingeben, (4) Speichern und (5) Einlesen von Registern. Man kann pro Datei maximal 150 Doppelstrings mit jeweils maximal 77 Zeichen Länge eingeben.
Vor dem Registermenü wird man gefragt, ob man ein „Buchregister" (Info = 4stellige Zahl) oder ein „sonstiges Register" (Info = 77stelliger String) anlegen will. Mit „Neustart", das jedoch alle Daten im Speicher löscht, kann man erneut zwischen Buch- und sonstigem Register wählen. Nur wenn Sie „Buchregister" wählen, werden später die numerischen Infos = Seitenzahlen richtig sortiert. Sollte ein Buchregister jedoch auch römische Seitenzahlen enthalten, so müssen Sie „sonstiges Register" wählen, denn römische Zahlen werden ohnehin nicht korrekt sortiert.
Der Registerredigierer ist im Gegensatz zu den anderen Programm-Modulen vergleichsweise narrensicher. Fremddateien (= Nicht-Registerdateien) werden zurückgewiesen. Ferner sind Eingabefehler praktisch ausgeschlossen.

Zeileneditierbefehle

Wenn man im Neueingabe- oder Änderungsmodus einen String neu eingibt oder einen alten String überschreibt, so kann man mit Ctrl-B zum String-Anfang und mit Ctrl-E zum String-Ende springen, mit Ctrl-D oder Del löschen, mit Ctrl-I einfügen (Einfüge-Modus aufheben mit Rechts- oder Linkspfeil), mit Ctrl-R oder Esc den alten String (nur im Änderungsmodus) aus dem Zwischenpuffer zurückholen, mit Ctrl-P den Stringrest bis zum Zeilenende abhacken, der dann allerdings auch aus dem Zwischenpuffer entfernt wird, usw.

Wenn Sie ein neues Register eingeben, so beginnen Sie, wie bereits erwähnt, mit einer leeren Registerdatendiskette, die keine anderen Dateien enthält. Dann kommen Sie auch nicht in Versuchung, irgendwelche Fremddateien einzulesen. Außerdem haben Sie einen besseren Überblick über die bereits angelegten Dateien. Erfin-

Leser werben Leser

»peeker« bietet Ihnen was!

Wer jetzt schenkt hat mehr von seinem Apple. Dafür schenkt »peeker« Ihnen etwas: Den praktischen Disk-Locher, der die Speicherkapazität Ihrer Disketten verdoppelt!

Sie wissen ja, wie gut der »peeker« Ihnen im täglichen Umgang mit Ihrem Apple behilflich ist. Und Sie brauchen Ihren »peeker« nicht mehr zu teilen oder auszuleihen.

Der blaue Disketten-Locher ist unser Geschenk an Sie für einen neuen »peeker«-Abonnenten. Denn wer einen Apple hat, der soll auch seinen »peeker« haben.

Bestellcoupon

Ich habe den neuen Abonnenten geworben und erhalte kostenlos den Disk-Locher.

Name, Vorname

Straße, Postfach

PLZ, Ort

X

Datum, Unterschrift

Ich bin der neue Abonnent. Bitte liefern Sie mir bis auf Widerruf, zumindest aber für 1 Jahr, »peeker« zum Jahresbezugspreis von DM 72,– (Ausland plus DM 18,– Porto) an folgende Anschrift:

Name, Vorname

Straße, Postfach

PLZ, Ort

X

Datum, Unterschrift

Gewünschte Zahlungsweise
☐ gegen Rechnung
☐ bargeldlos durch Bankeinzug

Konto-Nr. Bankleitzahl

Geldinstitut

Vertrauensgarantie:
Diese Bestellung kann ich innerhalb einer Woche bei Dr. Alfred Hüthig Verlag GmbH, Im Weiher 10, 6900 Heidelberg 1 widerrufen. Zur Wahrung der Frist genügt die rechtzeitige Absendung. Ich bestätige die Kenntnisnahme mit meiner Unterschrift:

X

2. Unterschrift

Peeker 7/86

Coupon ausschneiden und einsenden an:

»peeker«
Abonnementservice
Im Weiher 10
6900 Heidelberg 1

den Sie keine umständlichen Dateinamen! Nennen Sie die einzelnen Dateien
R1
R2
R3 usw., denn da jede Datei nur maximal 150 Doppelstrings enthält, entstehen z.B. bei einem 3000 Stichwörter umfassenden Register insgesamt 20 Dateien R1 ... R20.
Das Registerprogramm zeigt Ihnen vor dem Einlesen bzw. Speichern immer den Catalog der Diskette sowie speziell vor dem Speichern den Namen der zuletzt geladenen Datei, damit Sie nicht etwa R1 auf R3 speichern, womit R3 unwiderruflich gelöscht wäre.

2.5. Die anderen Hauptmodule

Während der Registerredigierer auch von „blutigen Laien" benutzt werden kann, setzen die anderen Module eine gewisse Vertrautheit mit dem Apple im allgemeinen und mit DOS 3.3 im besonderen voraus. Es wird deshalb auch nicht geprüft, ob etwa die zu sortierende Datei wirklich mit dem Registerprogramm erstellt wurde oder nicht. Würde man dem Registersortierer irgendeine Fremddatei „unterjubeln", so würde natürlich nur „Schrott" sortiert.

Registertester

Um sich trotzdem als Anfänger mit den einzelnen Modulen vertraut machen zu können, können wir Zufallswörter-Übungsdateien mit Hilfe des Registertesters erstellen. Hierzu booten wir die Registerprogrammdiskette und starten dann den Registertester (Menü-Option 5 im HELLO-Programm). Wir wählen dann
4 Testdateien mit
150 Doppelstrings,
die dann automatisch auf der Registerdatendiskette erzeugt werden und die Namen R1, R2, R3 und R4 erhalten.
Namenskonvention: Mit dem Registerpfleger oder hier mit dem Registertester angelegte Teildateien nennen wir zur Vermeidung von Verwechslungen R1, R2...Rn.

Registersortierer

Danach gelangen wir ohne unser Zutun in das HELLO-Programm zurück, von dem aus wir über Menü-Option 1 das Hauptprogramm starten (im Zwischenmenü Option deutsch statt amerikanisch wählen) und dann über die weitere Menü-Option 2 den Registersortierer starten, der bis zu 20 Teildateien „auf einen Schlag" sortieren kann. Wir geben die 4 Namen der Ausgangsdateien
R1, R2, R3 und R4
sowie als allgemeinen Zieldateinamen
S
ein. Es entstehen dann im Zuge der Sortierung aus den Dateien R1, R2, R3 und R4 die neuen Dateien
S.S1, S.S2, S.S3 und S.S4.
Die Suffixe „.S1" usw. werden automatisch an den allgemeinen Zieldateinamen hinzugefügt. Der Registersortierer hat also die *Ausgangsdateien* R1 usw. unversehrt gelassen (dies gilt für alle nachfolgenden Module) und daraus die sortierten Teildateien S.S1 usw. erzeugt, die allerdings im Zuge des nachfolgenden Mischens wieder gelöscht werden.
Wenn die Ausgangsdateien R1 usw. etwa die Hälfte der Datendiskette einnehmen, so müssen sie vor dem Mischen auf eine Zweitdiskette ausgelagert und dann auf der Mischdiskette gelöscht werden, damit genügend Platz für die Mischvorgänge bleibt.
Namenskonvention: Wenn nicht mehr als 20 Teildateien R1 usw. zu sortieren sind, so nennen wir das Präfix der Zieldatei S, womit aus R1 S.S1, aus R2 S.S2 usw. entstehen.

Registermischer

Wir haben nun 4 *in sich* sortierte Teildateien, doch unser Ziel ist eine komplett sortierte Gesamtdatei. Hierzu benötigen wir den Registermischer, den wir nach dem Verlassen des Registersortierers starten. Der Registermischer vereinigt jeweils zwei sortierte Teildateien zu einer sortierten Mischdatei. Wenn wir 4 Dateien S.S1, S.S2, S.S3 und S.S4 haben, so müssen im ersten Durchlauf
S.S1 und S.S2 zu A.M1 sowie
S.S3 und S.S4 zu A.M2
gemischt werden. Nach dem erfolgreichen Mischen werden S.S1 bis S.S4 gelöscht, so daß nur noch M1 und M2 übrigbleiben.
Im zweiten Durchlauf mischen wir
A.M1 und A.M2 zu B.M1
„A" und „B" sind hier die allgemeinen Zieldateinamen, die durch die Suffixe „.M1" usw. automatisch ergänzt werden.
Das Mischen ist dann besonders optimal, wenn die Anzahl der sortierten Teildateien eine Zweierpotenz darstellt (2, 4, 8, 16 usw.). Zur Veranschaulichung mischen wir den relativ ungünstigen Fall von 7 Ausgangsdateien:
S.S1 und S.S2 zu A.M1
S.S3 und S.S4 zu A.M2
S.S5 und S.S6 zu A.M3
S.S7 bleibt zunächst übrig. Jetzt mischen wir
A.M1 und A.M2 zu B.M1
A.M3 und S.S7 zu B.M2
und weiter geht's mit
B.M1 und B.M2 zu C.M1
Sortierte Teildateien (mit Suffix „.Sn") und Mischdateien (mit Suffix „.Mn") können in beliebiger Reihenfolge und beliebig oft gemischt werden.
Namenskonvention: Die Mischdateien erhalten das Präfix „A" usw. und ferner den automatischen Zusatz „.M1", „.M2" usw. Zum Schluß bleibt immer nur noch eine einzige Datei mit dem Suffix „.M1" übrig. Dies ist die fertig sortierte Enddatei, die ausgedruckt, umgewandelt oder geteilt werden kann.

Registerdrucker

Der Registerdrucker dient zum Ausdrukken der Enddatei mit Suffix „.M1". Wir müssen nach dem Programmstart spezifizieren, ob ein „Buchregister" oder ein „sonstiges Register" vorliegt, denn der Registerdrucker faßt während des Drukkens mehrere Infos zu demselben Stichwort zusammen (s.o). Zu Testzwecken können wir auf den Bildschirm ausdrucken (Slot 0).

Registerumwandler

Der Registerumwandler entspricht dem Registerdrucker, doch wird nicht auf den Drucker oder Bildschirm, sondern auf Diskette „gedruckt". Die so entstandene Datei mit Suffix „.U" kann direkt an die Druckerei gegeben werden oder auch von einem eigenen Textverarbeitungsprogramm nachbearbeitet, z.B. für die Setzerei codiert, werden.
Namenskonvention: Die umgewandelte Enddatei erhält das Suffix „.U". Sie kann von dem Registerprogramm niemals mehr eingelesen werden! Deshalb heben wir uns für die Registerpflege stets die Enddatei mit Suffix „.M1" auf.

Registerteiler

Wenn eine Enddatei mit Suffix „.M1" später durch neue Stichwörter usw. ergänzt werden soll, so muß sie zunächst wieder geteilt werden. Im Gegensatz zu den unsortierten Ausgangsdateien R1, R2 usw. erhalten wir jedoch jetzt sortierte Teildateien mit dem Suffix „.T1" usw., so daß wir bei Bedarf neue Stichwörter gleich an der richtigen Stelle einfügen können.
Diese Prozedur ist jedoch grundsätzlich nicht erforderlich, denn wir könnten z.B. eine Ergänzungsdatei – nennen wir sie E1 – anlegen. Es wird dann nur E1 zu S.S1 sortiert, und dann werden S.S1 und x.M1 zu einer neuen Enddatei y.M1 gemischt.
Namenskonvention: Geteilte Dateien erhalten das Suffix „.T1" usw.

Registerumdreher

Speziell bei zweisprachigen Wörterbüchern ist oft ein Umdrehen (z.B. deutsch-englisch in englisch-deutsch) gewünscht.

Nehmen wir an, die sortierte und gemischte Enddatei eines Wörterbuches laute W.M1.
Zunächst müssen wir diese Enddatei W.M1 teilen, denn Dateien mit mehr als 150 Doppelstrings lassen sich nicht sortieren. Es mögen dann die Dateien A.I1, A.I2 usw. entstehen, die wir nunmehr wieder wie unsortierte Ausgangsdateien (R1, R2 usw.) behandeln, also Sortieren, Mischen, fertig.
Namenskonvention: Umgedrehte Dateien erhalten das Suffix „.I1" usw. (I = invertiert).

3. Fehlermeldungen

Zu den wichtigsten Fehlermeldungen gehören:
9 = Diskette voll (neue Datendiskette nehmen)
6 = Datei nicht gefunden (Programmdiskette nicht eingelegt oder vertippt beim Eingeben der Dateinamen)
8 = I/O Fehler (Diskette wahrscheinlich defekt; Sie haben ja hoffentlich die Ausgangsdateien dupliziert?!)
77 = Speicher voll (theoretisch möglich, zumal bei 48K-DOS)
Vermeiden Sie Ctrl-Reset, weil dadurch erstens die Daten im Speicher und zweitens ggf. die Daten auf der Diskette zerstört werden können.

4. Geschwindigkeit

Ist das Registerprogramm schnell? Im Peeker, Heft 1/86, wurde ein Quicksort-Programm von Harald Grumser veröffentlicht, daß 1000 Zahlen in ca. 2s im RAM sortiert. Das Registerprogramm muß jedoch noch viel mehr tun:
Erstens müssen alle Dateien von Diskette geladen und wieder zurückgeschrieben werden.
Zweitens wird streng nach Duden und zudem voll durchsortiert. EDV-Sortierprogramme sortieren meistens nur nach ASCII und ferner oft nur z.B. bis zur 10. Stelle, also in keinem Fall bis etwa zur 77. Stelle bei langen Strings.
Drittens muß gemischt werden (Merge-Vorgang), was die meiste Zeit in Anspruch nimmt.
Viertens muß während des Ausdrucks oder Umwandelns noch untersortiert werden.

Test-Beispiel: Mit dem Registertester wurden 16 Ausgangsdateien (je 15 Sektoren) mit je 150 Doppelstrings erzeugt, also insgesamt 2400 Stichwörter (zusammen 224 Datensektoren = 56K). Für das Sortieren und Mischen wurden auf der MDB-Festplatte unter Einsatz der Accelerator IIe insgesamt einschließlich des Handlings (Modul-Wechsel, Eingabe der Dateinamen usw.) 12min und 25s benötigt. Und der Ausdruck auf den Bildschirm benötigte für die Enddatei (217 Sektoren) 1min und 13s. Bei Verwendung eines Disk-II-Laufwerks und ohne Accelerator sehen die Sortierzeiten noch erheblich trüber aus. Trotzdem halte ich das Registerprogramm für schnell. Da die Strings nämlich eine variable Länge von 1 bis 77 Zeichen haben können, kann man sich keine „schlauen" Diskettenalgorithmen ausdenken. Würde man nämlich hier eine Random-Access-Datei mit einer Record-Länge von 2 mal (77 + 1) = 156 Zeichen anlegen, so wäre die externe Datei mit 2400 mal 156 bereits 366K groß. Das Beispiel für externes Sortieren (QUICK.DISK in Peeker, 1/84, S. 14) hat ja bereits gezeigt, daß man auf einer RAM-Disk für eine Random-Datei mit 9999 Zahlen eine Stunde benötigt.
Man muß bei dem Registerprogramm, das mit sequentiellen Dateien arbeitet, bedenken, daß beispielsweise während des Mischens insgesamt 4 mal 56K = 224K (a) eingelesen und (b) gespeichert werden müssen, also insgesamt 448K. Hinzu kommen noch 2 mal 56K = 112K für das Sortieren und 1 mal 56K für das Ausdrucken, also zusammen 448K + 112k + 56K = 616K. Die meiste Zeit wird also beim Laden und Speichern der Daten „vertrödelt", während Codieren und Sortieren kaum ins Gewicht fallen.

dBase-Beispiel: Versuchen Sie einmal, mit dBase, Multiplan oder einem anderen Datenbankprogramm ein Register *extern nach Duden* zu sortieren. Wenn Sie bessere Zeiten ermitteln sollten, dann schreiben Sie an den Peeker.
Spaßeshalber habe ich für dBase II, Version 2.4, unter MBASIC, Version 5.27, eine Testdatei erzeugt, die 2400 Strings analog zum Registertester enthielt, jedoch nur Sortierwörter (39K), da hier Seiteninfos ohnehin nicht in Registerform sortierfähig wären. Das Sortieren dauerte dann unter CP/M 2.26 mit 6MHz-Z80-Karte auf einer 140K-Diskette über den Indizierungsbefehl (INDEX ON DATEI TO SORTFILE) 66min. Diese Sortierzeit muß als günstig angesehen werden, wenn man von den „kleinen" Nachteilen absieht, daß dBase II erstens Kleinbuchstaben, Umlaute und Sonderzeichen falsch sortiert und zweitens bei variablen Feldlängen (1-77 Zeichen für Wort und Info) praktisch auseinanderplatzen würde. Denn bei 2400 Doppelstrings würde man bei dBase (77+1) * 2 * 2400 * 2 = 748.800 Bytes an externem Speicher benötigen, und dann würde dBase II natürlich auch nicht mehr 2400 Doppelstrings in 66min sortieren.

aaa electronic gmbh
Neue Preise # 7

IMB-PC/XT-COMPATIBLE COMPUTER + ZUBEHÖR
Mainboard XT f. 16bit + 256K-RAM + 8 Slots, Leerkarte 79,--
Mainboard XT-640 K (on Bord) f. 16 Bit + 8 Stots Leerkarte 89,--
Mainboard TURBO XT-640 K (on Board) 4,7/8 MHz umsch. Leer 109,--
Disk Controller leer (f. 4 x 360KB-Drives) 34,--
Monochrome Leerkarte 49,--
Monochrome Graph. + par. Printer Leerkarte (Hercules comp.) 55,--
Color RGB + Video Graphik Leerkarte 55,--
Parallel Printer Leerkarte 29,--
512K-RAM Leerkarte 55,--
Disk-Multi I/O Leerkarte, (f. 2x360 K-Drive, 2xRS 232 1xpar. Printer Int., Game/Joystick Port, Real Time Clock/Kalender gepuft.) 59,--
384 K-Multifunktions Leerkarte, (RS 232, par. Printer Int., Game/Joystick Port, Real/time Clock/Kalender gepuffert) 55,--
Prototype/Lochraster Leerkarte, durchkontaktiert 64,--
Mainboard XT-256 f. 16 bit CPU, OK (bis 256K-on board, oder 640 K mit Addr. auf Karte aufr.) + Boot-ROM + 8 Slots best. + geprüft 189,--
Mainboard XT-640 f. 16 bit CPU, OK (bis 640 K-on board aufr.) gepr. 299,--
Mainboard XT-640 w.o. jedoch 256 K-RAM best. + gepr. 345,--
Mainboard XT-640 w.o. jedoch 640 K – vollbest. und geprüft 529,--
Mainboard TURBO XT-640 -4,7/6 MHz, OK, best. + gepr. 387,--
64KB-RAM-Chip-Aufrüstsatz (9 Stück) einzeln geprüft! 37,--
256 KB-RAM-Chip-Aufrüstung (9 Stück) einzeln geprüft! 97,--
Disk Controller gepr. (f. 4 x 360 KB-Drives)(Extra Kabel = 39,--) 94,--
Monochrome Karte geprüft 169,--
Monochrome Graph. + par. Printer Karte gepr. (Hercules comp.) 188,--
Color RGB + Video Graphikkarte gepr. 153,--
Parallel Printer-Karte gepr. o. Kabel (Extra Kabel = 49,-) 58,--
RS 232 Interface (2xRS 232, 1 davon bestückt) 83,--
512K-RAM-Karte gepr. (OKB) 117,--
384K-Multifunktionskarte, RS 232, par. Printer, Interf. Game/Joystick Port, Real Time Clock geputt., OK + RAM-Disk/Spooler/Uhr Softw. 245,--
Disk Multi I/O Karte (f. 2x360 K-Drive, 2xRS 232, 1xpar. Printer Int., Game/Joystick Port, Real Time Clock/Kalender geputt.) 243,--
Disk I/O w.o. sowie Color-Graphik/RGB+Composite+Monochrome TTL-Karte gepr. 489,--
EGA-Enhanced Graphic Adapter Karte, mit 256 K-RAM-Bildschirmspeicher, Monochrome-Color Grafik (640x350 Bildschirmpunkte – 16 Farben) voll bestückt + gepr. 977,--
Profi-Funktionstastatur ASCII + 15er Block ab 195,--/dt. ab 225,--
135W/150W-Netzteil mit eingebautem Ventilator, Kurzschlußf. 219,--/279,--
360KB-Diskdrive Slimline (Track Access Time <2 m S), R/W gepr. 295,--
Joystick f. IBM + Compatible 79,--
Maus incl. Software f. IBM + Compatible 277,--
PC/XT – KOMPLETT-SYSTEME EINGEBAUT UND GEPRÜFT
XT-640, 16 bit-Rechner (incl. Boot-Eprom) 256K-Ram im IBM-look-like Met.-Gehäuse + Profi-Fkt. Tast. + 135 W-Netzt. gepr. 999,--
XT-640, w.o. + Contr. + 360KB Drive + Color RGB + Vid. Graphik-Karte 1.4[illegible],--
XT-640, w.o. jedoch statt Color mit Monochrome Karte 1.499,--
XT-640, w.o. jedoch mit Monochrome-Graphik-Printer-Karte 1.525,--
XT-640, 256 K-RAM + 360 KB Drive + Disk Multi I/O + Color-Graphik-Karte 1.599,--
XT-640, 256 K-RAM + 360 KB Drive + Color RGB + Video-Graphik-Karte + 20 MB-NEC-Festplatte + DTC-Controller 3.275,--
Aufpreis für Turbo XT-640 -4,7/8 MHZ umschaltbares Board 55,--
Aufpreis für XT-640 M-board mit 640 K vollbestückt + gepr. 169,--
20 MB-NEC-Festplatte + DTC-Controller mit Kabel + Man., geprüft 1.885,--
PC 10-512 K + 1 Laufwerk + 20 MB-Festplatte 5749,--
640 K-Speichererweiterungs-Set für PC 10 279,--
PHOENIX GENIE 16 C -PC-compatibel

APPLE-BUS COMPUTERPLATINEN + PERIPHERIE
Motherboard II 48K leer, durchkont. mit Lötstoppplack 57,--
Motherboard II 64 K leer 88,--
Motherboard II 64 K+Z 80 CPU, leer 95,--
Motherboard IIe, 64 K leer 98,--
M-board II 48K, gesockelt, vollbest. + gepr. 337,--
Motherboard II 64K gesockelt, vollbest. +gepr. 375,--
M-board II 64K + Z 80 CPU, gesockelt, v. best. + gepr. 378,--
M-board IIe 64K vollbestückt + gepr. 389,--
APOLLO II Kunststoff-Leergehäuse 85,--
APOLLO II Gehäuse incl. Funkt.-Tast. 217,--
Superstarkes Netzteil – Kurzschlußfest 5A=115,-- b.z.w. 7A= 145,--
Leerplatine: 16K-RAM, Z80/CPU-, Controller DOS 3.3, Parallel-Druckerkarte, Ser. Int. V24, 80Z/24Z-Karte, PAL-Karte je 18,95
Entwicklungslochraster-Leerplatine durchkontaktiert 27,--
16K-Speicher-Karte geprüft 55,--
APPLE orig. 16K-Language Karte gepr. 98,--
Z-80 CPU-Karte geprüft (CP/M Vol. I.+II Manual dafür = 69,--) 64,--
Controller DOS 3.3-Karte geprüft 63,--
Auto-Controller DOS 3.2/3.3-Karte gepr. 97,--
Erphi AFDC 2 Controller + Autopatch Softw. + Handbuch 278,--
Parallel-Drucker-Karte geprüft (Extra Kabel = 45,--) 74,--
Par.-Drucker-Grappler comp. Leerplatine 27,--
Par.-Drucker Karte-Grappler comp. geprüft (Extra Kabel=45,--) 169,--
8bit par. Int. + 64K-Buffer + Kabel gepr. (Grappler comp.) 299,--
Seriell-Interface-Karte V24 geprüft 88,--
Super-Seriell Leerkarte 34,--
Super-Seriell-Interface Karte gepr. 169,--
80-Zeichen/24-Zeilen-Karte geprüft 119,--
80Z/24Z-Karte + Softswitch-Schalter Leerplatine 24,--
80Z/24Z-Karte + Softswitch-Schalter, gest. Scharf geprüft 127,--
80Z/24Z-Softwitch Karte gepr. für IIe 49,--
80Z/24Z+64K-Ram+Softsw. Leerkarte f. IIe 27,--
80Z/24Z+64K-Ram+Softsw.-Karte best.+gepr. f. IIe 87,--
IEEE-488 Int. Karte gepr. (Extra Kabel=45,--) 297,--
PAL-Modulator/Color-Karte geprüft + UHF Modulator 99,--
RGB-Color Karte gepr. 129,--
128K-Speicher-Leerkarte 29,--
128K-Speicher-Leerkarte + Software + Manual 99,--
128K-Speicher-Karte + Software geprüft + Manual 255,--
EPROM-Burner (2716/32/64) geprüft 109,--
6522-VIA Karte geprüft 135,--
Clock Karte geprüft + Software + Manual 94,--
Speech Karte geprüft + Software + Manual 53,--
Wild Karte geprüft + Software + Manual (nicht f. IIe) 59,--
AD/DA-8Bit Karte gepr. 375,--
UHF/TV-Modulator universell 27,50
Lüfter anclipsbar (220 V) 79,--
Dataphon s21d Akustikkoppler-FTZ mit RS 232/V24 Anschl. 275,--
APOLLO//(48K) + UHF-Mod. + Gr/Ki vollbest. + gepr. 617,--
APOLLO//(48K) + Disk II F + Contr. + 12"-Monitor 1.285,--
APOLLO//A (48K) + Gr/Ki + 15er-Tastatur vollbest. + gepr. 677,--
APOLLO/A (48K) + Disk II F + Contr. + 12"-Mon. 1.345,--
APOLLO// ASKF (48K) vollbest. + gepr. Sep., Keyboard + Gr/Ki + 15er-Block mit Funkt.-Tasten im „IBM-look like" Kunststoff-Gehäuse 835,--
APOLLO// ASKF (48K) w.o. + Disk II F + Contr. 1.179,--
Apollo II ASFK (48K) w.o. + 2xDisk II-F + Contr. + 12" Mon. 1.787,--
Aufpreis f.o. Systeme von 48K zu 64K 55,--
Aufpreis von 48K zu 64K + Z80 CPU 89,--
Apollo IIe (64K) vollbest. und geprüft 725,--
Apollo IIe (128K) + 80Z/24Z-Karte geprüft 799,--
Applle IIe (64K) + 15er-Tastatur vollbestückt 785,--
Apollo IIe ASKF (64 K) vollbestückt und geprüft ab 876,--
Apollo IIe (64K) + Disk II F + 12" Monitor 1.409,--
APPLE IIe 128K + 80Z/24Z Karte + Softw. 1.998,--
Disk-Drives voll APPLE II, 100%ig compatibel + R/W geprüft + Track Ø Sensor
Disk//+Contr. + Kabel DOS 3.3 (Siemens) im Geh. 529,--
Disk//(Siemens) im Geh. + Kabel 467,--
Disk//F-HS (High Speed 1/2 Höhe) im Gehäuse + Contr. + Kabel 379,--
Disk//F-HS (High Speed 1/2 Höhe) im Geh. + Kabel (Track Ac. <2m S) 318,--
Disk//F-HS-80 Track im Gehäuse + Kabel I. orig. Contr. geeignet 389,--
Disk//F-HS-80 w.o. + Contr. + DOS 3.3 + CP/M Patch-Softw. 489,--
Erphi DuoDisk 1,2 MByte im Gehäuse + Erphi-AFDC 3-Contr. 1.195,--
APPLESOFT + Turorial + Reference Manual engl. 75,--
APPLE-DOS 3.3 Manual engl. = 25,--/Handbuch dt. 45,--
APPLE-Pascal Reference + Operating Manual engl. 48,--
APPLE-Pascal Reference Hb. dt. 49,--
APPLE-Pascal Language Hb. dt. 39,--
APPLEII-Pascal Hb. dt. (te-wi) 59,--
APPLE-Fortran Manual engl. 69,--
CP/M-Softcard Vol. I + II, Manual engl. 69,--
APPLE II + IIe-Anwenderhandbuch deutsch (te-wi) 66,--
Z80 Karte gepr. f. APPLE/c + RAM-Disk-Software + Hb. 489,--
Macintosh Umrüstung von 128K auf 512K 567,--
EPSON FX 85F/T+, 8bit/parallel 160 Z/S. NLQ + IBM Komp. 1.275,--
EPSON FX 105+, 8bit/parallel 15" sonst w.o. 1.696,--
EPSON Traktoraufsatz für FX 80 + FX 85 129,50
APPLE/EPSON-Drucker Graphic-Interface + Kabel 169,--
MX 80/82, FX/RX 80 Spezialfarbband-Kassette 15,95
BROTHER-Typenradr: HR 15XL 8bit/par. neuestes Mod. 1.387,--

Wir führen verschiedene Monitore von Zenith, Phillips und Sanyo mit Video – TTL (IBM Komp.) oder Coloreingang. – Bitte Preisliste anfordern! –

Disketten in Box + Aufkleber I. Wahl, 10er Pack/100er Pack./ Stückpreis:

5¼"	BASF	3M SCOTCH	FUJI	DATA MAGNETICS	NEUTRAL
1X, SS/SD	3,49/ 3,29				
1D, SS/DD	3,59/ 3,39	3,38/ 2,98	4,27/ 3,97	2,56/2,46	2,25/2,15
2D, DS/DD	4,59/ 4,29	4,58/ 3,98	5,87/ 5,47	3,26/3,06	2,75/2,55
1D/96TPI	4,49/ 4,19	4,88/ 4,68			
2D/96TPI	4,99/ 4,79	5,48/ 5,18	7,87/ 7,67		
2HD/1,2MB	9,69/ 9,39	9,98/ 8,18	12,87/12,57		
1X, Hard	4,29/ 4,19				
8"					
1X, SS/SD	4,49/ 4,39	6,68/ 6,48	6,97/ 6,67		
1D, SS/DD	4,79/ 4,59	6,58/ 6,28			
2D, DS/DD	7,09/ 6,89	9,88/ 9,58	10,17/ 9,77		
1X, Hard	4,49/ 4,39	6,68/ 6,48			
3½"					
Einseitig	7,29/ 6,89	6,78/ 6,08	7,67/ 7,17		
Zweiseitig	11,59/11,19	9,28/ 8,48	11,17/ 9,97		

Das sind Preise!

3M-Scotch 5¼" Flüssig-Reinigungsset (2 Disk.) zweiseitig 48,95
5¼" Disketten-Archivbox für 10 Disks (1 St./10 St.) 4,75/43,50
5¼" Disk-Karteikasten Kunststoff (ca. 80 Disk) 26,50
5¼" Disk-Karteikasten Kunststoff (ca. 80 Disk) Rauchglas 39,95
5¼" Disk-K.K.Rauchglas, abschließb. (f. ca. 80-100 Disks) 47,50
2000 Bl. Tabpapier (24 cm x 12" einf.) weiß oder grün/weiß perf. 45,--
4000 Et.-Aufkl. doppelt. (107 x 36 mm auf 240 x 12" perf. Trägerp.) 66,50
4000 Etik.-Aufkl. einreih. (107 x 36 mm auf 125 x 12" perf. Trägerp.) 65,--

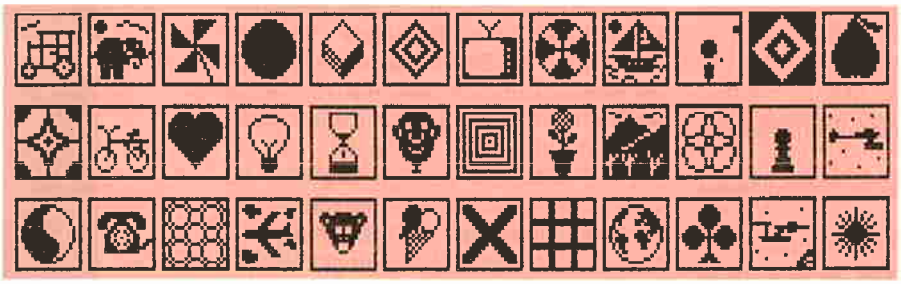
OSZILLOSCOPE HAMEG ab Lager
Bei Vorauszahlung frei Empfangsstation unversichert in der BRD, ausgenommen Papier und Etiketten, sonst N.N. + V.S. ab DM 30,--
Öffnungszeiten: Mo, Di, Do, Fr v. 10-18 h, Mi u. Sa v. 10-14 h, Ia Sa v. 10-18 h.
Telef. Best.: Mo, Di, Do, Fr von 10-19 h, Mi u. Sa wie Öffnungszeiten.
GEWÄHRLEISTUNG: 6 Monate, auf alle bei uns gekauften Geräte, durch unsere eigene Service-Werkstatt.
REPARATUREN an Apple + Compatible Geräte + Zubehör führt unser Spezialistenteam garantiert zuverlässig + besonders kostengünstig aus.
Sprechen sie mit uns. Kostenvorschlag auf Wunsch!

aaa electronic gmbh
Telex: 0772642 aaa-d
Habsburgerstraße 134
7800 FREIBURG, Tel. (0761) 276864
Bauelemente – Bausätze – µP's
Meßgerate – Zubehor – Fachliteratur
Fachgeschäft fur Elektronik + Mikrocomputer

Mousory

Memory-Spiel mit der Apple-Maus

von Wolfgang Landgraf



Hinweis: Das Spiel läuft auf allen Apple-II-Typen und kann auch ohne Apple-Maus verwendet werden.

Mit dem kleinen Neffen meiner Freundin spielte ich vor einiger Zeit Memory – ein Spiel, bei dem man unter einer Anzahl umgedrehter Karten Bildpaare finden muß. Das ewige Drehen der Kärtchen, das erneute Mischen und Auslegen nach einer beendeten Runde wurde auf die Dauer recht mühsam, und so entstand die Idee, das Spiel auf den Computer zu übertragen und die unangenehmen Aufgaben dem Rechner zu überlassen. Das vorliegende Memory-Spiel ist das Ergebnis einiger arbeitsamer Tage. Ich hoffe, damit auch den Peeker-Lesern eine Freude gemacht zu haben.

1. Programm-Module

MOUSORY läuft auf den Rechnern Apple II+, IIe und IIc. Es ist mit und ohne Apple-Maus zu spielen, wobei die Maus nicht unbedingt in Slot 4 beheimatet sein muß. MOUSORY besteht aus folgenden Programmen:

MOUSORY: Hello-Programm, das die Grafik MOUSORY.BILD und dann entweder MOUSORY.DEMO oder MOUSORY.GAME lädt.
MOUSORY.BILD: Nur auf Sammeldisk #19 enthalten.
MOUSORY.DEMO: Demo-Programm für diejenigen, die sich mit den Spielregeln vertraut machen wollen.
MOUSORY.GAME: Dies ist das eigentliche Programm, das später direkt mit RUN MOUSORY.GAME gestartet werden kann. Das gelistete MOUSORY.GAME enthält REM-Zeilen, die auf der Sammeldisk #19 zur Vermeidung von Speicherkonflikten bereits entfernt worden sind.
MOUSORY.BELL: erzeugt Töne zur akustischen Untermalung.
MOUSORY.SET1, **MOUSORY.SET2** und **MOUSORY.SET3:** Dies sind drei Shapetabellen. Eine Tabelle reicht für das Spiel jedoch aus, so daß MOUSORY.SET2 und MOUSORY.SET3 nur auf der Sammeldisk #19 enthalten sind.

2. Spielregeln

Man kann das Spiel zu zweit oder auch allein gegen den Computer spielen, wobei das Gedächtnis des Rechners in Schritten von 25% zwischen 0% und 100% wählbar ist. Spielt man mit der Maus, so wird der Cursor unter das umzudrehende Bild gesteuert, und der gedrückte Mousebutton dreht das ausgewählte Kärtchen. Danach sucht man sich ein zweites Bild aus und dreht es auf die gleiche Weise. Gibt es eine Übereinstimmung, so verschwinden beide Bildchen vom Schirm, der Spieler erhält zwei Punkte und ist erneut an der Reihe. Sind die Karten unterschiedlich, so ist der Gegner an der Reihe.
Wer keine Maus besitzt, steuert den Cursor beim Apple II+ mit den I-J-K-M-Tasten. Apple IIe und IIc bieten zusätzlich die Pfeiltasten. Das Umdrehen der Karten geschieht dann durch Druck auf die ESC- oder 0-Taste.

Zum Zug des Computers sei noch gesagt, daß er nur Karten kennt, die schon einmal umgedreht wurden, und daß der erste Zug immer zufällig geschieht. Doch schon beim 75%igen Gedächtnis ist es sehr schwer, dem Kollegen Computer das Wasser abzugraben. Nun kann ich nur noch viel Spaß wünschen, auf daß der Mensch der bessere MOUSORY-Spieler werde.

3. Änderungsmöglichkeiten

All denen, die sich die Mühe machen, die Applesoft-Listings MOUSORY.GAME usw. abzutippen, sei geraten, die Remarks wegzulassen, da das Programm mit seinen Variablen sonst in die benutzte HGR-Seite hineinragt.

Shapes: Wer weitere Kartensätze außer dem gelisteten MOUSORY.SET1 und den zwei weiteren Shape-Tabellen auf der Sammeldiskette erstellen möchte, sollte die Größe von 21 * 21 Bildpunkten nicht überschreiten, wobei das Shape in der Bildmitte seinen Ausgang nehmen sollte:

```
..................... 01
..................... 02 usw. bis
.........*........... 11
..................... 12 usw. bis
..................... 21
```

Der dreieckige Auswahlcursor, der unter den Karten wandert, sollte nicht höher als 5 Bildpunkte sein. Die Deckkarte ist in jeder Tabelle das erste Shape und der Auswahlcursor das letzte. Insgesamt gibt es 36 Kartenshapes (Shape-Nummern 2-37) sowie die Deckkarte (Shape-Nummer 1) und den Auswahlcursor (Shape-Nummer 38).

Anm.d.Red.: Die Shapes wurden von Herrn Landgraf mit einem eigenen Shape-Editor erstellt, der bei entsprechender Nachfrage im Peeker veröffentlicht werden kann.

Bei nur *einem* Kartensatz müssen im Programm einige Zeilen geändert oder weggelassen werden. Die Variable ZF bestimmt den zu ladenden Zeichensatz. Sie wird in den Zeilen 620-625 definiert und in Speicherstelle 793 (dezimal) gepokt. In den Zeilen 835-850 wird dann nach Einlesen dieses Wertes der entsprechende Kartensatz geladen. Die Zeilen 620, 835, 845 und 850 entfallen, und die Zeilen 625 und 840 müssen wie folgt geändert werden:

625 RUN
840 PRINT D$"BLOAD MOUSORY.SET1"

Das Programm nutzt zur Erkennung des Computertyps das MACHID-Byte in der Speicherstelle $BF98 der ProDOS-Global-Page. Dies hat jedoch keinen Einfluß auf die Funktionsfähigkeit unter DOS 3.3, so daß das Programm direkt von der Peeker-Sammeldisk #19 (im DOS-Format) gestartet werden kann. Man kann die Programm-Module, die ursprünglich unter ProDOS erstellt wurden, aber auch mit CONVERT oder DOSTOPRO auf eine ProDOS-Diskette übertragen.

Kurzhinweise

1. Zweck:
Memory-Spiel mit Hires-Spielkarten;
2. Konfiguration:
Apple II+/e/c; optional mit Apple-Maus; DOS 3.3 oder ProDOS;
3. Test:
RUN MOUSORY;
4. Sammeldisk:
MOUSORY
MOUSORY.DEMO
MOUSORY.GAME
MOUSORY.BELL
MOUSORY.SET1
MOUSORY.SET2
MOUSORY.SET3
MOUSORY.BILD

MOUSORY (Startprogramm)

```
1ØØ HOME : REM MOUSORY
1Ø5 D$ = CHR$ (4)
11Ø HGR2 : HGR
115 PRINT D$"BLOADMOUSORY.SET2,A$6ØØØ"
12Ø POKE 232,Ø: POKE 233,96: HCOLOR= Ø: ROT= Ø: SCALE= 1
125 PRINT D$"BLOADMOUSORY.BILD,A$4ØØØ"
13Ø POKE 23Ø,64
135 POKE - 163Ø2,Ø: POKE - 16299,Ø
14Ø P = PEEK ( - 16384):P1 = PEEK (794)
145 IF P > 127 THEN 23Ø
15Ø IF P1 THEN FOR D = 1 TO 2Ø
155 I = INT ( RND (1) * 8) + 1
16Ø ON I GOSUB 19Ø,195,2ØØ,2Ø5,21Ø,215,22Ø,225
165 XDRAW Z AT X,Y: XDRAW 1 AT X,Y
17Ø FOR J = 1 TO 12ØØ: NEXT
175 XDRAW 1 AT X,Y: XDRAW Z AT X,Y
18Ø IF P1 THEN NEXT : GOTO 26Ø
185 GOTO 14Ø
19Ø Z = 18:X = 261:Y = 142: RETURN
195 Z = 31:X = 216:Y = 43: RETURN
2ØØ Z = 12:X = 54:Y = 16: RETURN
2Ø5 Z = 17:X = 91:Y = 134: RETURN
21Ø Z = 14:X = 135:Y = 2Ø: RETURN
215 Z = 3Ø:X = 17:Y = 13Ø: RETURN
22Ø Z = 8:X = 14Ø:Y = 135: RETURN
225 Z = 15:X = 183:Y = 16: RETURN
23Ø POKE - 16368,Ø: TEXT : HOME : POKE 233,Ø
235 VTAB 14: HTAB 6: PRINT "Das Demo kann durch
    Tastendruck"
24Ø VTAB 16: HTAB 12: PRINT "abgebrochen werden."
245 VTAB 11: HTAB 11: PRINT "Demo,Spiel (1/2) : ";: GET
    A$:A = VAL (A$)
25Ø IF A < 1 OR A > 2 THEN 245
255 PRINT : ON A GOTO 26Ø,265
26Ø PRINT D$"RUN MOUSORY.DEMO"
265 PRINT D$"RUN MOUSORY.GAME"
```

MOUSORY.DEMO (Demoprogramm)

```
1ØØ REM MOUSORY.DEMO
1Ø5 REM ** Wolfgang Landgraf, Bielefeld 1985 **
11Ø REM ** Kein Spiel, nur Demonstration **
115 HOME
12Ø GOSUB 465
125 PE = PEEK (49Ø48):N2$ = "SPIELER":SZ = 2
13Ø N1$ = "APPLE "
135 IF PE < 128 THEN N1$ = N1$ + "II+": GOTO 15Ø
14Ø IF PE < 18Ø THEN N1$ = N1$ + "//E"
145 IF PE > = 18Ø THEN N1$ = N1$ + "//C"
15Ø GOSUB 455
155 I = 1: INVERSE : HTAB 3: PRINT N1$: NORMAL : VTAB 23:
    HTAB LEN (N1$) + 4: PRINT S(1): VTAB 23: HTAB 36 - LEN
    (N2$): PRINT N2$" "S(2)
16Ø GOTO 265
165 GOSUB 455
17Ø I = 2: HTAB 3: PRINT N1$: VTAB 23: HTAB LEN (N1$) + 4:
    PRINT S(1): VTAB 23: HTAB 36 - LEN (N2$): INVERSE :
    PRINT N2$;: NORMAL : PRINT " "S(2)
175 GOTO 265
18Ø IF V1 = V2 THEN S(I) = S(I) + 2:SH(I) = 1: POKE
    769,2Ø: POKE 781,1Ø: GOTO 2ØØ
185 SH(I) = Ø
19Ø FOR J = 1 TO 2ØØØ: NEXT : XDRAW B1 AT X1,Y1: XDRAW B2
    AT X2,Y2: XDRAW 1 AT X1,Y1: XDRAW 1 AT X2,Y2
195 GOTO 2Ø5
2ØØ XDRAW B1 AT X1,Y1: CALL 768: XDRAW B2 AT X2,Y2: CALL
    768:A(P3,P4) = Ø:A(P5,P6) = Ø: POKE 769,3Ø: POKE
    781,2Ø
2Ø5 IF S(1) + S(2) = 72 THEN 37Ø
21Ø IF I = 1 AND SH(I) THEN 15Ø
215 IF I = 2 AND SH(I) THEN 165
22Ø IF I = 1 AND NOT SH(I) THEN P5 = Ø:P6 = Ø:P3 = Ø:P4 =
    Ø: GOTO 165
225 IF I = 2 AND NOT SH(I) THEN P5 = Ø:P6 = Ø:P3 = Ø:P4 =
    Ø: GOTO 15Ø
23Ø IF A(P1,P2) = Ø THEN Z = Z - 1: RETURN
235 XDRAW 1 AT X,Y - 14: XDRAW A(P1,P2) + 1 AT X,Y - 14
24Ø B(P1,P2) = A(P1,P2)
245 IF Z = 1 THEN V1 = B(P1,P2):X1 = X:Y1 = Y - 14:B1 =
    B(P1,P2) + 1:P3 = P1:P4 = P2
25Ø IF Z = 2 THEN V2 = B(P1,P2):X2 = X:Y2 = Y - 14:B2 =
    B(P1,P2) + 1:P5 = P1:P6 = P2
255 IF P3 = P5 AND P4 = P6 THEN Z = 1: CALL 768: XDRAW 1
    AT X2,Y2: XDRAW B2 AT X2,Y2
26Ø RETURN
265 CB = 5
27Ø IF PEEK ( - 16384) > 128 THEN 45Ø
275 Z = 1
28Ø P1 = INT ( RND (1) * 12) + 1:P2 = INT ( RND (1) * 6) +
    1
285 X = 13 + (P1 - 1) * 23:Y = 27 + (P2 - 1) * 26
29Ø IF X = X1 AND Y - 14 = Y1 THEN 28Ø
295 IF A(P1,P2) = Ø THEN 28Ø
3ØØ GOSUB 23Ø
3Ø5 IF Z = 2 THEN 315
31Ø GOTO 32Ø
315 GOTO 18Ø
```

```
320 FOR K = 1 TO 12: FOR L = 1 TO 6
325 IF P1 = K AND P2 = L THEN 350
330 IF B(K,L) = 0 THEN 350
335 IF P1 < > K AND P2 < > L AND B(P1,P2) = B(K,L) THEN P1
    = K:P2 = L:K = 0:L = 0:Z = 2: GOTO 360
340 IF P1 < > K AND P2 = L AND B(P1,P2) = B(K,L) THEN P1 =
    K:P2 = L:K = 0:L = 0:Z = 2: GOTO 360
345 IF P1 = K AND P2 < > L AND B(P1,P2) = B(K,L) THEN P1 =
    K:P2 = L:K = 0:L = 0:Z = 2: GOTO 360
350 NEXT : NEXT
355 Z = 2: GOTO 280
360 X = 13 + (P1 - 1) * 23:Y = 27 + (P2 - 1) * 26: GOSUB
    230
365 GOTO 180
370 FOR I = 1 TO 1000: NEXT
375 TEXT : HOME : INVERSE
380 T1$ = "DER GEWINNER IST"
385 FOR I = 1 TO 3: PRINT T$;: NEXT
390 VTAB 2: PRINT TAB( 21 - LEN (T1$) / 2);T1$: NORMAL
395 IF S(1) > S(2) THEN VTAB 11: HTAB 12: PRINT "Mit
    "S(1)" Punkten !": VTAB 6: HTAB 21 - ( LEN (N1$) / 2):
    FLASH : PRINT N1$:PL = 1:PP = 2: NORMAL : GOTO 415
400 IF S(2) > S(1) THEN VTAB 11: HTAB 12: PRINT "Mit
    "S(2)" Punkten !": VTAB 6: HTAB 21 - ( LEN (N2$) / 2):
    FLASH : PRINT N2$: NORMAL : GOTO 415
405 NORMAL
410 VTAB 11: HTAB 4: PRINT "Beide hatten dieselbe
    Punktzahl !":PL = 1
415 IF PL THEN T1$ = "Punktzahl " + N2$ + " : ": VTAB 15:
    HTAB 1: PRINT TAB( 20 - LEN (T1$) / 2);T1$;S(2): IF PP
    THEN 425
420 T1$ = "Punktzahl " + N1$ + " : ": VTAB 17: HTAB 1:
    PRINT TAB( 20 - LEN (T1$) / 2);T1$;S(1)
425 VTAB 23: HTAB 1: PRINT "Nochmal (J/N) : ";: FOR T = 1
    TO 500: NEXT : PRINT "J"
430 ZF = ZF + 1: IF ZF = 3 THEN ZF = 0
435 POKE 793,ZF
440 IF PEEK ( - 16384) > 127 THEN POKE - 16368,0: GOSUB
    455: GOTO 450
445 PRINT D$"RUN MOUSORY"
450 PRINT D$"RUN MOUSORY.GAME"
455 IF ER THEN ER = 0: CALL 768
460 VTAB 23: HTAB 1: CALL - 868: RETURN
465 D$ = CHR$ (4)
470 ZF = PEEK (793): IF ZF = 1 OR ZF = 2 THEN 480
475 PRINT D$"BLOADMOUSORY.SET1": GOTO 490
480 IF ZF = 1 THEN PRINT D$"BLOADMOUSORY.SET2": GOTO 490
485 IF ZF = 2 THEN PRINT D$"BLOADMOUSORY.SET3"
490 PRINT D$"BLOADMOUSORY.BELL"
495 POKE 232,0: POKE 233,64: POKE 794,1
500 HGR : HCOLOR= 3: SCALE= 1: ROT= 0:CO = INT ( RND (1) *
    2)
505 IF CO THEN HPLOT 0,0: CALL 62454: HCOLOR= 0
510 HPLOT 0 + CO,0 + CO TO 0 + CO,159 - CO TO 279 - CO,159
    - CO TO 279 - CO,0 + CO TO 0 + CO,0 + CO
515 FOR P1 = 13 TO 277 STEP 23: FOR P2 = 13 TO 157 STEP 26
520 XDRAW 1 AT P1,P2
525 NEXT : NEXT
530 T$ = "                                        "
535 VTAB 21: INVERSE : PRINT T$: VTAB 21: HTAB 13: PRINT
    "MOUSE -- MEMORY"
540 NORMAL
545 GOSUB 455
550 VTAB 23: HTAB 8: PRINT "Bitte einen Moment Geduld"
555 DIM A(12,6): DIM B(12,6)
560 FOR I = 1 TO 36
565 FOR J = 1 TO 2
570 X = INT ( RND (1) * 12) + 1:Y = INT ( RND (1) * 6) + 1
575 IF A(X,Y) < > 0 THEN 570
580 A(X,Y) = I
585 NEXT : NEXT
590 X = 13:Y = 27:FL = 1:P1 = 1:P2 = 1
595 RETURN
```

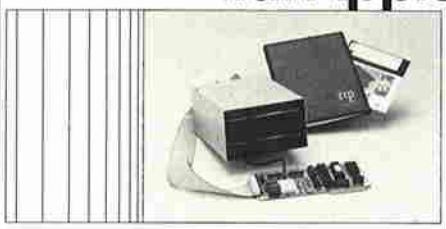
MOUSORY.GAME (Eigentliches Programm)

(Auf Peeker-Sammeldisk #19 ohne REMs!)

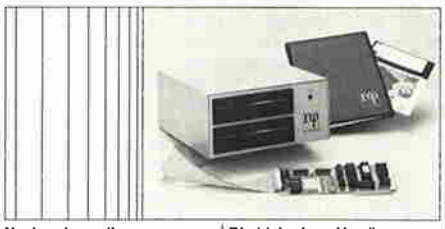
```
100 REM MOUSORY.GAME - Listing mit REMs
105 REM ** Wolfgang Landgraf, Bielefeld 1985 **
110 REM ** Bitte alle REM's weglassen **
115 HOME
120 GOSUB 830
125 GOSUB 645
130 REM ** Eingaben zum Spiel **
135 PRINT "Mouse, Tastatur (1/2) :";: GOSUB 660: GOSUB 645
140 IF I < 1 OR I > 2 THEN ER = 1: GOSUB 645
145 EG = I - 1: IF NOT EG THEN GOSUB 970
150 PRINT "Ein Spieler, zwei Spieler (1/2) : ";: GOSUB 660
155 IF I < 1 AND I > 2 THEN ER = 1: GOSUB 645: GOTO 150
160 IF I = 1 THEN SZ = 1
165 GOSUB 645
170 INPUT "Name Spieler 1 : ";N1$
175 IF N1$ = "" THEN ER = 1: GOTO 165
180 IF SZ THEN PE = PEEK (49048): GOTO 205
185 GOSUB 645
190 INPUT "Name Spieler 2 : ";N2$
195 IF N2$ = "" OR N2$ = N1$ THEN ER = 1: GOTO 185
200 GOTO 240
205 N2$ = "APPLE "
210 IF PE < 128 THEN N2$ = N2$ + "II+": GOTO 225
215 IF PE < 180 THEN N2$ = N2$ + "//E"
220 IF PE > = 180 THEN N2$ = N2$ + "//C"
225 GOSUB 645
230 PRINT N2$":0,25,50,75,100% (1/2/3/4/5):";: GOSUB 660
235 CS = I: IF CS < 1 OR CS > 5 THEN ER = 1: GOTO 225
240 GOSUB 645
245 PRINT "Anfang: "N1$" (1) "N2$" (2) : ";: GOSUB 660
250 IF I < 1 AND I > 2 THEN ER = 1: GOTO 240
255 IF I = 2 AND SZ = 1 THEN SZ = 2
260 IF I = 2 THEN N3$ = N1$:N1$ = N2$:N2$ = N3$:N3$ = ""
265 GOSUB 910
270 GOSUB 645
275 REM ** 1. Spieler **
280 I = 1: INVERSE : HTAB 3: PRINT N1$: NORMAL : VTAB 23:
    HTAB LEN (N1$) + 4: PRINT S(1): VTAB 23: HTAB 36 - LEN
    (N2$): PRINT N2$" "S(2)
285 IF SZ = 2 THEN 420
290 GOTO 670
295 GOSUB 645
300 REM ** 2. Spieler **
305 I = 2: HTAB 3: PRINT N1$: VTAB 23: HTAB LEN (N1$) + 4:
    PRINT S(1): VTAB 23: HTAB 36 - LEN (N2$): INVERSE :
    PRINT N2$;: NORMAL : PRINT " "S(2)
310 IF SZ = 1 THEN 420
315 GOTO 670
320 REM ** Drehen der Karten **
325 IF V1 = V2 THEN S(I) = S(I) + 2:SH(I) = 1: POKE
    769,20: POKE 781,10: GOTO 345
330 SH(I) = 0
335 FOR J = 1 TO 2000: NEXT : XDRAW B1 AT X1,Y1: XDRAW B2
    AT X2,Y2: XDRAW 1 AT X1,Y1: XDRAW 1 AT X2,Y2
340 GOTO 350
345 XDRAW B1 AT X1,Y1: CALL 768: XDRAW B2 AT X2,Y2: CALL
    768:A(P3,P4) = 0:A(P5,P6) = 0: POKE 769,30: POKE
    781,20
350 IF S(1) + S(2) = 72 THEN 540
355 IF I = 1 AND SH(I) THEN 270
360 IF I = 2 AND SH(I) THEN 295
365 IF I = 1 AND NOT SH(I) THEN P5 = 0:P6 = 0:P3 = 0:P4 =
    0: GOTO 295
370 IF I = 2 AND NOT SH(I) THEN P5 = 0:P6 = 0:P3 = 0:P4 =
    0: GOTO 270
375 REM ** Zuordnung der Bilder auf Versuche **
380 IF A(P1,P2) = 0 THEN Z = Z - 1: RETURN
385 XDRAW 1 AT X,Y - 14: XDRAW A(P1,P2) + 1 AT X,Y - 14
390 B(P1,P2) = A(P1,P2)
395 IF Z = 1 THEN V1 = B(P1,P2):X1 = X:Y1 = Y - 14:B1 =
    B(P1,P2) + 1:P3 = P1:P4 = P2
400 IF Z = 2 THEN V2 = B(P1,P2):X2 = X:Y2 = Y - 14:B2 =
    B(P1,P2) + 1:P5 = P1:P6 = P2
405 IF P3 = P5 AND P4 = P6 THEN Z = 1: CALL 768: XDRAW 1
    AT X2,Y2: XDRAW B2 AT X2,Y2
410 RETURN
415 REM ** Subroutine Computerzug **
420 BB = INT ( RND (1) * 4) + 1:CB = CS - BB
425 XDRAW 38 AT X,Y
430 IF CB > 0 THEN Z = 1: GOTO 440
435 FOR Z = 1 TO 2
440 P1 = INT ( RND (1) * 12) + 1:P2 = INT ( RND (1) * 6) +
    1
445 X = 13 + (P1 - 1) * 23:Y = 27 + (P2 - 1) * 26
450 IF X = X1 AND Y - 14 = Y1 THEN 440
455 IF A(P1,P2) = 0 THEN 440
460 GOSUB 380
465 IF Z = 2 THEN 480
470 IF CB > 0 THEN 485
475 NEXT
480 XDRAW 38 AT X,Y: GOTO 325
485 FOR K = 1 TO 12: FOR L = 1 TO 6
490 IF P1 = K AND P2 = L THEN 515
495 IF B(K,L) = 0 THEN 515
500 IF P1 < > K AND P2 < > L AND B(P1,P2) = B(K,L) THEN P1
    = K:P2 = L:K = 0:L = 0:Z = 2: GOTO 525
```

```
505 IF P1 < > K AND P2 = L AND B(P1,P2) = B(K,L) THEN P1 =
    K:P2 = L:K = Ø:L = Ø:Z = 2: GOTO 525
510 IF P1 = K AND P2 < > L AND B(P1,P2) = B(K,L) THEN P1 =
    K:P2 = L:K = Ø:L = Ø:Z = 2: GOTO 525
515 NEXT : NEXT
520 Z = 2: GOTO 440
525 X = 13 + (P1 - 1) * 23:Y = 27 + (P2 - 1) * 26: GOSUB
    380
530 XDRAW 38 AT X,Y: GOTO 325
535 REM ** Spielende **
540 FOR I = 1 TO 1000: NEXT
545 TEXT : HOME : INVERSE : IF EG THEN 565
550 PRINT D$"PR#"N: PRINT CHR$ (Ø)
555 PRINT D$"PR#Ø"
560 PRINT D$"IN#Ø"
565 T1$ = "DER GEWINNER IST"
570 FOR I = 1 TO 3: PRINT T$;: NEXT
575 VTAB 2: PRINT TAB( 21 - LEN (T1$) / 2);T1$: NORMAL
580 IF S(1) > S(2) THEN VTAB 11: HTAB 12: PRINT "Mit
    "S(1)" Punkten !": VTAB 6: HTAB 21 - ( LEN (N1$) / 2):
    FLASH : PRINT N1$:PL = 1:PP = 2: NORMAL : GOTO 600
585 IF S(2) > S(1) THEN VTAB 11: HTAB 12: PRINT "Mit
    "S(2)" Punkten !": VTAB 6: HTAB 21 - ( LEN (N2$) / 2):
    FLASH : PRINT N2$: NORMAL : GOTO 600
590 NORMAL
595 VTAB 11: HTAB 4: PRINT "Beide hatten dieselbe
    Punktzahl !":PL = 1
600 IF PL THEN T1$ = "Punktzahl " + N2$ + " : ": VTAB 15:
    HTAB 1: PRINT TAB( 20 - LEN (T1$) / 2);T1$;S(2): IF PP
    THEN 610
605 T1$ = "Punktzahl " + N1$ + " : ": VTAB 17: HTAB 1:
    PRINT TAB( 20 - LEN (T1$) / 2);T1$;S(1)
610 VTAB 23: HTAB 1: PRINT "Nochmal (J/N) : ";: GET X$: IF
    X$ < > "J" AND X$ < > "j" AND X$ < > "N" AND X$ < >
    "n" THEN 600
615 IF X$ = "N" OR X$ = "n" THEN 630
620 ZF = ZF + 1: IF ZF = 3 THEN ZF = Ø
625 POKE 793,ZF: RUN
630 HOME : VTAB 22: NEW
635 END
640 REM ** Texte am unteren Bildrand **
645 IF ER THEN ER = Ø: CALL 768
650 VTAB 23: HTAB 1: CALL - 868: RETURN
655 REM ** Eingabe der Werte **
660 GET I$: PRINT I$:I = VAL (I$): RETURN
665 REM ** Eingabe mit Tastatur **
670 FOR Z = 1 TO 2
675 IF NOT EG THEN GOSUB 745
680 VTAB 10: GET E$
685 XDRAW 38 AT X,Y
690 IF E$ = CHR$ (27) OR E$ = "Ø" THEN 725
695 IF E$ = "I" OR E$ = "i" OR E$ = CHR$ (11) THEN Y = Y -
    26:P2 = P2 - 1: IF Y < 27 THEN Y = Y + 26: CALL 768:P2
    = P2 + 1:FL = 1
700 IF E$ = "M" OR E$ = "m" OR E$ = CHR$ (10) THEN Y = Y +
    26:P2 = P2 + 1: IF Y > 157 THEN Y = Y - 26: CALL
    768:P2 = P2 - 1:FL = 1
705 IF E$ = "J" OR E$ = "j" OR E$ = CHR$ (8) THEN X = X -
    23:P1 = P1 - 1: IF X < 13 THEN X = X + 23: CALL 768:P1
    = P1 + 1:FL = 1
710 IF E$ = "K" OR E$ = "k" OR E$ = CHR$ (21) THEN X = X +
    23:P1 = P1 + 1: IF X > 266 THEN X = X - 23: CALL
    768:P1 = P1 - 1:FL = 1
715 IF FL THEN XDRAW 38 AT X,Y: GOTO 675
720 XDRAW 38 AT X,Y:FL = 1: GOTO 675
725 XDRAW 38 AT X,Y
730 GOSUB 380
735 NEXT : GOTO 325
740 REM ** Eingabe mit Mouse **
745 IF OK THEN 765
750 PRINT D$"PR#"N: PRINT CHR$ (1)
755 PRINT D$"PR#Ø"
760 PRINT D$"IN#"N:OK = 1
765 IF PEEK ( - 16384) > 127 THEN POKE - 16368,Ø
770 XDRAW 38 AT X,Y
775 VTAB 10: HTAB 1: INPUT "";X,Y,S
780 X = INT (X / 93 * 4)
785 Y = INT (Y / 51)
790 IF X > 11 THEN X = 11
795 IF Y > 5 THEN Y = 5
800 P1 = X + 1:P2 = Y + 1:X = 13 + X * 23:Y = 27 + Y * 26
805 XDRAW 38 AT X,Y
810 IF S < Ø THEN POKE - 16368,Ø
815 IF S < 3 THEN POP :S = 3: XDRAW 38 AT X,Y: GOTO 725
820 GOTO 770
825 REM ** Subroutine Initialisierung **
830 D$ = CHR$ (4)
835 ZF = PEEK (793): IF ZF = 1 OR ZF = 2 THEN 845
840 PRINT D$"BLOADMOUSORY.SET1": GOTO 855
845 IF ZF = 1 THEN PRINT D$"BLOADMOUSORY.SET2": GOTO 855
850 IF ZF = 2 THEN PRINT D$"BLOADMOUSORY.SET3"
855 PRINT D$"BLOADMOUSORY.BELL"
860 POKE 232,Ø: POKE 233,64
865 HGR : HCOLOR= 3: SCALE= 1: ROT= Ø:CO = INT ( RND (1) *
    2)
870 IF CO THEN HPLOT Ø,Ø: CALL 62454: HCOLOR= Ø
875 HPLOT Ø + CO,Ø + CO TO Ø + CO,159 - CO TO 279 - CO,159
    - CO TO 279 - CO,Ø + CO TO Ø + CO,Ø + CO
880 FOR P1 = 13 TO 277 STEP 23: FOR P2 = 13 TO 157 STEP 26
885 XDRAW 1 AT P1,P2
890 NEXT : NEXT
895 T$ = "                                        ": REM **
    40 Spaces **
900 VTAB 21: INVERSE : PRINT T$: VTAB 21: HTAB 13: PRINT
    "MOUSE -- MEMORY"
905 NORMAL : RETURN
910 GOSUB 645
915 VTAB 23: HTAB 8: PRINT "Bitte einen Moment Geduld"
920 DIM A(12,6): DIM B(12,6)
925 FOR I = 1 TO 36
930 FOR J = 1 TO 2
935 X = INT ( RND (1) * 12) + 1:Y = INT ( RND (1) * 6) + 1
940 IF A(X,Y) < > Ø THEN 935
945 A(X,Y) = I
950 NEXT : NEXT
955 X = 13:Y = 27: XDRAW 38 AT X,Y:FL = 1:P1 = 1:P2 = 1
960 RETURN
965 REM ** Initialisierung der Mouse **
970 L1 = 49420:L2 = 49659
975 FOR M = 1 TO 7: IF PEEK (L1) = 32 AND PEEK (L2) = 214
    THEN N = M:M = 9
980 L2 = L2 + 256:L1 = L1 + 256
985 NEXT M
990 IF M > 8 THEN RETURN
995 POP :ER = 1: GOSUB 645: PRINT "          Du hast keine
    Mouse !!": FOR M = 1 TO 2000: NEXT : GOSUB 645:EG = 1:
    GOTO 150
```


Anschlußfertig für Apple® II, //e...



erphi
Doppellaufwerk DL 280
2 Laufwerke
mit je 640 kB formatiert
im Gehäuse incl.
erphi
Autopatch-Controller AFDC 3
Handbuch und Diskette mit
Dienstprogrammen
Verkaufspreis incl. MwSt.
DM 1.298,–



erphi
Floppy-Subsystem FSS 280
für kommerzielle Anwendungen
2 Laufwerke
mit je 640 kB formatiert
im Gehäuse mit eigener
Stromversorgung incl.
erphi
Autopatch-Controller AFDC 3
Handbuch und Diskette mit
Dienstprogrammen
Verkaufspreis incl. MwSt.
DM 1.698,–

Noch universeller und noch komfortabler...

Betriebssysteme
DOS 3.3
DiversiDOS 2-c, 4-c,
ProDOS 1.0.1, 1.1.0, 1.1.1
Pascal 1.1, 1.2
CP/M 2.20, 2.23, 2.26

SLOT-Unabhängig

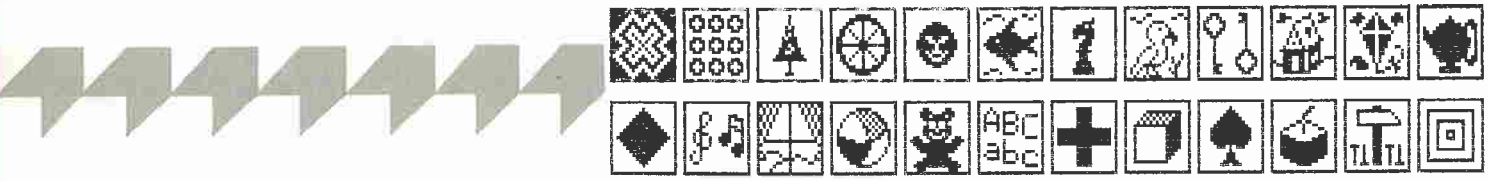
Die bisherigen Vorzüge bleiben erhalten, wie

Autopatch-Boot
automatische Erkennung und Erweiterung der Betriebssysteme während des Bootvorgangs (Betriebssystem auf der Boot-Diskette bleibt unverändert)

Originalsystem-Boot
Von herkömmlichen Apple®-Disketten (unabhängig vom Laufwerksformat) weiterhin möglich

Problemloses Übertragen
herkömmlicher Apple®-Software auf Disketten höherer Kapazität durch SIM 35-Hilfsprogramm (simuliert 35-Spur-Laufwerk unabhängig vom tatsächlichen Laufwerksformat)

erphi electronic
GmbH
Dammweg 3
D-8011 Großhelfendorf
Telefon (08095) 441
Telex 528021 erphi d



MOUSORY.BELL (Ton-Routine)

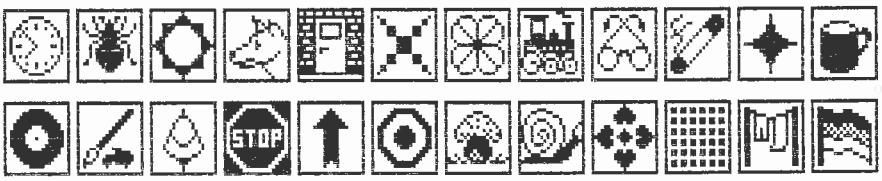
```
BSAVE MOUSORY.BELL,A$0300,L$19

$0300: A9 1E 85 3C A9 14 20 A8
$0308: FC 8D 30 C0 A9 14 20 A8
$0310: FC 8D 30 C0 C6 3C D0 EC
$0318: 60 00 00 00 00 00 00 00
```

MOUSORY.SET1 (Hires-Shapes)

```
BSAVE MOUSORY.SET1,A$4000,L1611
```

Die Shape-Tabellen MOUSORY.SET2 und MOUSORY.SET3 befinden sich auf der Sammel-disk #19

```
$4000: 26 00 4E 00 01 01 E9 01
$4008: 89 02 08 03 B3 03 22 04
$4010: AB 04 1D 05 C1 05 70 06
$4018: B6 06 7F 07 33 08 1B 09
$4020: 8F 09 45 0A 9E 0A FD 0A
$4028: 9C 0B 21 0C 9E 0C 7A 0D
$4030: 62 0E B5 0E 17 0F A0 0F
$4038: 88 10 33 11 C7 11 46 12
$4040: C1 12 6E 13 F0 13 7C 14
$4048: 04 15 73 15 06 16 C0 DB
$4050: DB DB 40 18 40 18 40 03
$4058: 2D 2D 2D 2D 2D 2D 2D 2D
$4060:  2D 2D 36 36 36 36 36 36
$4068:  36 36 36 36 3F 3F 3F 3F
$4070: 3F 3F 3F 3F 3F 3F 24 24
$4078: 24 24 24 24 24 24 24 6C
$4080: B9 17 96 05 28 28 28 28
$4088: 68 F1 1E 1E 1E 1E 1E 1E
$4090: 1E 96 96 96 05 28 28 28
$4098: 28 28 28 28 28 28 28 28
$40A0: 28 28 28 28 28 F8 F3 1E
$40A8: 1E 1E 1E 1E 1E 1E 1E 1E
$40B0: 1E 1E 1E 04 40 03 05 28
$40B8: 28 28 28 28 28 28 28 28
$40C0: 68 49 91 2A B0 B2 B2 B2
$40C8: B2 BA DF 05 28 28 20 40
$40D0: 03 17 17 17 17 17 17 DF
$40D8: 05 28 28 28 28 28 28 28
$40E0: 28 20 40 03 17 9F F1 1E
$40E8: 1E 1E 1E 1E 1E 1E 1E 1E
$40F0: DF 05 28 28 28 28 28 28
$40F8: 28 28 28 28 28 28 28 20
$4100: 00 D8 D8 D8 D8 D8 48 D8
$4108: 48 D8 C8 2D 2D 2D 2D 2D
$4110: 2D 2D 2D 2D 2D 3E 1B 1B
$4118: 1B 1B 1B 1B 1B 1B 1B 33
$4120: 0D 09 09 29 09 09 09 09
$4128: 09 09 3E 1B 1B 1B 1B 1B
$4130: 1B 3B 3F 1B 33 0D 2D 09
$4138: 09 09 09 09 09 09 09 3E
$4140: 3B 3F 3F 3F 3F 3F 1F 1B
$4148: 1B 33 0D 09 09 09 0D 09
$4150: 29 09 29 09 3E 1B 1F 1B
$4158: 1F 1B 3B 1B 1B 1B 33 0D
$4160: 09 0D 09 0D 09 29 09 29
$4168: 09 3E 1B 3F 3F 3F 3F 3F
$4170: 3F 3F 1B 33 0D 09 0D 09
$4178: 0D 09 29 09 29 09 3E 1B
$4180: 1F 1B 3F 1F 3B 1B 3B 1B
$4188: 33 0D 09 0D 09 0D 09 29
$4190: 2D 2D 09 3E 1B 3F 3B 1F
$4198: 1B 3B 1B 3B 1B 33 0D 29
$41A0: 2D 09 0D 09 2D 09 29 0D
$41A8: 3E 3B 1B 1F 3B 3F 3F 3F
$41B0: 1F 3F 33 0D 0D 09 0D 09
$41B8: 09 2D 09 29 0D 3E 1B 3F
$41C0: 3B 1F 1B 1B 3B 1F 3F 33
$41C8: 0D 29 2D 09 09 09 09 2D
$41D0: 0D 09 3E 1B 1B 1B 1B 1B
$41D8: 1B 1B 1B 1B 33 2D 2D 2D
$41E0: 2D 2D 2D 2D 2D 2D 2D 06
$41E8: 00 49 49 19 36 25 B4 1A
$41F0: BE 27 05 40 18 08 20 3F
$41F8: 38 F7 1E 07 38 3F 3F 1E
$4200: BE 5E 6F 30 36 15 36 2E
$4208: 25 24 CD 35 36 2D 24 2C
$4210: 40 18 D8 40 03 3F 3F 37
$4218: 36 36 35 36 04 40 2D 2D
$4220: 3C 3F 3F 24 2C 2D 2D 15
$4228: 15 2D 24 3F 24 35 25 56
$4230: 36 4D 05 F8 DB DB 12 2E
$4238: 3C 38 3F 3F 2C 2D B5 D2
$4240: 36 04 40 18 40 18 40 07
$4248: 38 3F FF DB 24 24 24 2C
$4250: 2D 2D 2D 2D 2D 2D 2D 2D
$4258: 2D 35 36 36 36 36 36 36
$4260: 36 36 36 3E 3F 3F 3F 3F
$4268: 3F 3F 3F 3F 3F 9C 24 24
$4270: 24 24 24 24 0D 40 28 05
$4278: 20 07 38 17 2E 25 6F 49
$4280: 29 05 28 15 2D 0E AD 04
$4288: 00 49 49 49 21 24 24 24
$4290: 24 3C 3F 3F 3F 3F 3F 3F
$4298: 3F 3F 3F 37 36 36 36 36
$42A0: 36 36 36 36 36 2E 2D 2D
$42A8: 2D 2D 2D 2D 2D 2D 2D 25
$42B0: 24 24 24 24 DF DB DB DB
$42B8: DB 1B 40 18 40 18 40 03
$42C0: 36 36 36 25 24 24 15 36
$42C8: 36 25 24 15 36 25 15 05
$42D0: 20 24 24 2C 2D 2D F5 3F
$42D8: 3F 2E 2D 1E 3F 2E 1E 96
$42E0: 28 2D 2D 35 36 36 3E 20
$42E8: 24 3C 36 3E 20 3C 3E F8
$42F0: 36 36 36 3F 3F 3F 0C 2D
$42F8: 2D 3C 3F 0C 2D 3C 0C 0D
$4300: 18 A8 AA AE AE AE 26 00
$4308: DB DB DB 23 24 24 24 24
$4310: 2C 2D 2D 2D 2D 2D 2D 2D
$4318: 2D 2D 35 36 36 36 36 36
$4320: 36 36 36 36 3E 3F 3F 3F
$4328: 3F 3F 3F 3F 3F 3F 27 24
$4330: 24 24 24 45 24 05 20 05
$4338: 28 28 28 05 28 2D AD AD
$4340: 15 15 76 36 36 17 BE 17
$4348: 17 BF 3F 3F 1C E7 07 38
$4350: 38 08 1C AC 24 24 05 20
$4358: 2D 2D 2D 2D 2D 2D 1C 3F
$4360: 3F 3F 3F 3F 0C 2D 2D 2D
$4368: 2D DC 3F 3F 96 DB 2A 2D
$4370: 2D 2D 2D 2D 15 3F 3F 3F
$4378: 3F 3F 3F 37 2D 2D 2D 2D
$4380: 2D 2D 35 3F 3F 3F 3F 3F
$4388: 3F 37 2D 2D 2D 2D 2D 2D
$4390: 35 3F 3F 3F 3F 3F 3F 37
$4398: 2D 2D 2D 2D 2D 3E 3F
$43A0: 3F 3F 3F 3F AF 2D 2D 2D
$43A8: 2D 2D 1E 3F 3E 3C 3E 3C
$43B0: E6 27 00 DB DB DB 23 24
$43B8: 24 24 24 2C 2D 2D 2D 2D
$43C0: 2D 2D 2D 2D 2D 35 36 36
$43C8: 36 36 36 36 16 B4 36 36
$43D0: 3F 3F 3F 3F 3F 3F 3F 3F
$43D8: 3F 3F 24 24 24 24 6C 49
$43E0: 49 92 92 89 05 28 28 28
$43E8: 28 28 28 18 24 3C 38 38
$43F0: 38 38 E0 1E 1E 1E 1E 1E
$43F8: 1E 36 36 15 15 15 15 15
$4400: 25 24 0C 05 28 28 28 F8
$4408: DB DB 1B 15 15 15 15 16
$4410: 1C 07 38 38 68 49 92 31
$4418: 2E 20 2C 30 2E 20 2C 30
$4420: 26 00 DB DB DB 23 24 24
$4428: 24 24 2C 2D 2D 2D 2D 2D
$4430: 2D 2D 2D 2D 35 36 36 36
$4438: 36 36 36 36 36 36 3E 3F
$4440: 3F 3F 3F 3F 3F 3F 3F 3F
$4448: 27 24 24 24 24 4C 49 49
$4450: C9 DB DB 2B 28 28 28 28
$4458: 28 28 28 A8 15 15 15 15
$4460: 15 15 15 17 17 17 17 17
$4468: 17 17 17 07 38 38 38 38
$4470: 38 38 20 05 28 28 28 28
$4478: 28 28 A8 15 15 15 15 15
$4480: 15 17 17 17 17 17 17 17
$4488: 1C 1C 1C 1C 1C 1C 0D 28
$4490: 28 28 28 A8 15 15 15 1E
$4498: 1E 1E 1E 1C 1C 1C 2C 28
$44A0: 28 A8 15 15 1E 1E 1E 07
$44A8: 38 20 00 DB DB DB 23 24
$44B0: 24 24 24 2C 2D 2D 2D 2D
$44B8: 2D 2D 2D 2D 2D 35 36 36
$44C0: 36 36 36 36 36 36 36 3E
$44C8: 3F 3F 3F 3F 3F 3F 3F 3F
$44D0: 3F 27 24 24 24 24 45 92
$44D8: 92 CA 22 24 24 24 24 24
$44E0: 2D 2D 2D 2D 2D 2D 2D 2D
$44E8: 36 36 36 36 36 3E 3F 3F
$44F0: 3F 3F 3F 3F 3F 24 40 18
$44F8: 40 18 40 03 4D 49 49 49
$4500: 2B 36 36 36 36 3E 0D 25
$4508: 3C 04 28 26 20 27 28 26
$4510: 38 C4 DB 1B 2C 28 28 F8
$4518: 93 E3 1C 04 00 DB DB DB
$4520: 23 24 24 24 24 2C 2D 2D
$4528: 2D 2D 2D 2D 2D 2D 2D 35
$4530: 36 36 36 36 36 36 36 36
$4538: 36 3E 3F 3F 3F 3F 3F 3F
$4540: 3F 3F 3F 27 24 24 24 24
$4548: 4C 24 05 20 05 28 28 28
$4550: 05 28 2D 75 75 0E 0E 76
$4558: 36 36 17 BE 17 17 BF 3F
$4560: 3F 1C E7 1C 1C 3C 20 4D
$4568: 49 09 20 05 28 F8
$4570: DB AB 15 B6 52 BE BE 1E
$4578: 4D 49 19 1C 07 20 04 40
$4580: 03 2D 28 05 28 B0 92 3A
$4588: 38 38 DF 18 E7 E7 1C 96
$4590: 92 05 28 28 3C 17 24 07
$4598: 38 36 36 25 24 04 40 49
$45A0: 51 2C 38 27 27 2D 2E 2C
$45A8: 6E 49 92 3A 3E 3E 6F AA
$45B0: 25 27 25 96 92 D2 DB 23
$45B8: 37 27 1E 27 25 2D 1C 24
$45C0: 00 DB DB DB 23 24 24 24
$45C8: 24 2C 2D 2D 2D 2D 2D 2D
$45D0: 2D 2D 2D B5 B4 36 36 36
$45D8: 36 36 36 36 36 3E 3F 3F
$45E0: 3F 3F 3F 3F 3F 3F 3F 27
$45E8: 24 24 24 24 8D 2D 2E 2E
$45F0: 2E 2E 2D 2D 25 25 25 25
$45F8: 2D D6 9B 3E 1F 38 3F E7
$4600: 0D 0D 0D 0C 3F 3F 3F 3F
$4608: 1C 2D 6D 21 24 24 24 24
$4610: AC 15 15 15 15 15 15 3E
$4618: 3F 3F 27 04 20 20 8D 32
$4620: AD 37 2D DF FF 03 40 18
$4628: 40 18 17 17 17 1E 1E 1E
$4630: 1E 2D 2D 2D 24 24 BC 36
$4638: 3E 24 17 3E 07 40 18 40
$4640: 18 98 12 2D 38 3F 2C 2D
$4648: 1C B7 92 92 92 D2 2B 0E
$4650: 2D 70 49 49 49 01 28 0E
$4658: 05 B0 9B BF DF 1B 07 38
$4660: F7 DC E7 4D 49 49 49 49
$4668: 01 40 18 40 18 01 20 00
$4670: 10 27 2D 36 3F 27 2C 28
$4678: 75 36 17 3F 56 3E 0D 3E
$4680: 37 2D 3E 77 3E 3F 3F 3F
$4688: 3F 27 24 24 24 24 24 24
$4690: 24 24 24 2C 2D 2D 2D 2D
$4698: 2D 2D 2D 2D 2D 35 36 3E
$46A0: 37 2D 36 36 36 36 FE DB
$46A8: 03 B0 4A 49 08 36 36 3E
$46B0: 3F 3F 3F 3F 04 00 DB DB
$46B8: DB 23 24 24 24 24 2C 2D
$46C0: 2D 2D 2D 2D 2D 2D 2D 2D
$46C8: 35 36 36 36 36 36 36 36
$46D0: 36 36 3E 3F 3F 3F 3F 3F
$46D8: 3F 3F 3F 3F 27 24 24 24
$46E0: 24 AD 15 15 15 15 15 15
$46E8: 15 0D 0C 0C 0C 0C 0C 0C
$46F0: 0C 0C 04 38 38 38 38 38
$46F8: 38 38 38 F8 1E 1E 1E 1E
$4700: 1E 1E 1E 1E 24 24 24 24
$4708: 2D 2D 2D F5 3F 3F 37 2D
$4710: 2D 1E 3F 1E 30 2D 1E 37
$4718: 96 D2 36 36 36 2E 24 24
$4720: 24 15 36 36 2E 24 24 15
$4728: 36 2E 24 15 2E 4D 29 2D
$4730: 2D 2D 3C 3F 3F 2F 28 2D
$4738: 2D 3C 3F 67 2D 25 3F 0C
$4740: 25 04 40 18 3C 07 28 25
$4748: 3F E7 2D 2D 3C 3F 3F 1C
$4750: 2D 2D 2D 3C 3F 3F 3F 96
$4758: 9B 17 17 17 17 17 15 15
$4760: 15 15 15 05 28 28 28 28
$4768: E0 1C 1C 1C 1E 30 37 37
$4770: 37 77 15 30 35 35 05 20
$4778: 25 25 E5 27 27 04 00 DB
$4780: DB DB 83 24 24 24 24 24
$4788: 2D 2D 2D 2D 2D 2D 2D 2D
$4790: 2D 2D 36 36 36 36 36 36
$4798: 36 36 36 36 3F 3F 3F 3F
$47A0: 3F 3F 3F 3F 3F 3F 24 24
$47A8: 24 24 64 49 49 49 92 92
$47B0: 12 28 1F 1E 3F 1C 1C E4
$47B8: 24 2C 20 0C 64 05 28 20
$47C0: 05 28 2C 16 15 96 92 92
$47C8: 92 12 2D 0C 05 20 05 20
$47D0: 24 07 20 07 20 07 38 B0
$47D8: 92 92 92 D2 1B 04 28 35
$47E0: 25 15 2C 3C 3F 3F 17 17
$47E8: 3F 1C 2D 2D 38 3F E7 2D
$47F0: 2D 2D 2D 2D 3C 3F 3F
$47F8: 3F 3F 3F 27 2D 2D 2D 2D
$4800: 2D 2D 3C 3F 3F 3F 3F 3F
$4808: 67 2D 2D 2D 2D 3C 3F
$4810: 3F 3F 3F 2F 28 2D 2D 2D
$4818: 25 3F 3F 3F 3F 0C 2D 2D
$4820: 2D 1C 3F 3F 05 28 E5 6F
$4828: 01 28 15 2D 2E F5 3F 38
$4830: 07 20 00 D8 D8 D8 D8 D8
$4838: 48 D8 48 D8 C8 2D 2D 2D
$4840: 2D 2D 2D 2D 2D 2D 2D 3E
$4848: 3F 3F 1B 1B 3B 1B 1B 3B
$4850: 3F 37 2D 2D 0D 09 09 0D
$4858: 09 09 2D 2D 3E 3F 1F 1B
$4860: 1B 3B 1B 1B 1B 3F 37 2D
$4868: 0D 09 09 29 2D 09 09 09
$4870: 2D 3E 1B 1B 1B 1B 3F 1F
$4878: 1B 1B 1B 33 0D 09 09 09
$4880: 2D 29 0D 09 09 09 3E 1B
$4888: 1B 1B 3B 1F 3F 1B 1B 1B
$4890: 33 0D 09 09 2D 0D 09 2D
$4898: 0D 09 09 3E 1B 3B 3F 1F
$48A0: 3B 1B 3F 3F 1B 33 2D 2D
$48A8: 2D 09 29 2D 09 29 2D 2D
$48B0: 3E 1B 3B 3F 1F 3B 1B 3F
$48B8: 3F 1B 33 0D 09 09 2D 0D
$48C0: 09 2D 0D 09 09 3E 1B 1B
$48C8: 1B 3B 1F 3F 1B 1B 1B 33
$48D0: 0D 09 09 09 2D 29 0D 09
$48D8: 09 09 3E 1B 1B 1B 1B 3F
$48E0: 1F 1B 1B 1B 33 2D 0D 09
$48E8: 09 29 2D 09 09 09 2D 3E
$48F0: 3F 1F 1B 1B 3B 1B 1B 1B
$48F8: 3F 37 2D 2D 0D 09 09 0D
$4900: 09 09 2D 2D 3E 3F 3F 1B
$4908: 1B 3B 1B 1B 3B 3F 37 2D
$4910: 2D 2D 2D 2D 2D 2D 2D 2D
$4918: 2D 06 00 DB DB DB 23 24
$4920: 24 24 24 2C 2D 2D 2D 2D
$4928: 2D 2D 2D 2D 2D 35 36 36
$4930: 36 36 36 36 36 36 36 3E
$4938: 3F 3F 3F 3F 3F 3F 3F 3F
$4940: 3F 27 24 24 24 24 4D 92
$4948: 92 0A 2D 05 28 20 3C 38
$4950: 38 F7 1E 36 15 4D 49 49
$4958: 49 49 20 3C 38 38 F7 1E
$4960: 36 0E 15 2D 05 E0 C0 3A
$4968: E8 C3 F8 DB 93 13 05 20
$4970: C5 20 24 07 38 38 10 8D
$4978: 09 05 28 96 DA 2D 2D 2E
$4980: 2C 28 38 6F 21 B7 D2 28
$4988: 2D 2D 96 9B 24 20 00 40
$4990: 18 40 18 40 18 92 12 07
$4998: 68 05 28 AD 15 36 1E 1E
$49A0: 17 17 17 17 07 38 38 38
$49A8: 38 38 20 64 0C 2D 96 92
$49B0: 92 52 91 1C 1C 1C 1C 1C
$49B8: 1C 1C 1C 24 24 05 28 28
$49C0: 28 2D 15 15 05 28 28 2D
$49C8: 0E 15 15 36 36 1E 1E 1E
$49D0: 1E 1E 1E 1E 27 6F 2C 2C
$49D8: 2C 2C 2C 2C 24 24 27 27
$49E0: 3F F7 1E 1C 1C 3F 37 37
$49E8: 37 36 2E 2E 2E 2E 2E 2E
$49F0: 05 60 3F 1C 2D 2D 2C 2C
$49F8: 2C 24 27 3F 17 3E 27 27
$4A00: 3F 3E 36 35 35 2D 2D 05
$4A08: 20 25 3D 38 BE 3F 27 27
$4A10: 17 15 15 2D 25 40 18 40
$4A18: 18 40 03 3F 3F 3F 3F 3F
$4A20: 37 36 36 36 36 36 36 36
$4A28: 36 36 2E 2D 2D 2D 2D 2D
$4A30: 2D 2D 2D 2D 25 24 24 24
$4A38: 24 24 24 24 24 24 3C 1F
$4A40: FD 3F 3F 04 00 40 18 40
$4A48: 18 80 2D DF F7 F7 1E 36
$4A50: 76 AE 15 15 4D 29 28 28
$4A58: 20 05 20 24 07 38 38 96
$4A60: 92 92 DB 22 2C 28 F8 1B
$4A68: 15 96 1A 2D 2D 3E 3F 37
$4A70: 2D 2D 1E 3F 16 3F 3F 3F
$4A78: 3F 27 24 24 24 24 24 24
$4A80: 24 24 24 2C 2D 2D 2D 2D
$4A88: 2D 2D 2D 2D 2D 35 36 36
$4A90: 36 36 36 36 36 36 36 3E
$4A98: 3F 3F 3F 3F 27 00 40 18
$4AA0: 40 18 40 03 2D 2D 36 36
$4AA8: 17 17 17 17 07 38 38 38
```



Hobby

```
$4AB0: 20 24 2C 2D 96 92 2D 1E
$4AB8: 56 1F 1E 1E 1E 36 36 2D
$4AC0: 2D 2D 25 3F 3F 3F 2C 2D
$4AC8: 2D B5 21 24 E4 1C 1F B0
$4AD0: FE DB DB 1B 24 24 24 24
$4AD8: 24 24 24 2D 2D 2D 2D 2D
$4AE0: 2D 2D 2D 2D 2D 36 36 36
$4AE8: 36 36 36 36 36 36 36 3F
$4AF0: 3F 3F 3F 3F 3F 3F 3F 3F
$4AF8: 3F 24 24 24 00 40 18 40
$4B00: 18 40 03 AD 75 15 AE 36
$4B08: 17 36 17 BE 17 B8 3F 07
$4B10: 38 1C 3C 20 3C 20 2C 20
$4B18: 05 28 28 0C 96 92 92 92
$4B20: 52 BA 2D 04 40 18 40 18
$4B28: D8 93 2A 28 6D 3E 1E 3F
$4B30: 3F 8D 2D 05 28 44 3C 3F
$4B38: 3F 3F 36 07 20 4D 01 40
$4B40: 03 24 35 FE 24 5F 37 AE
$4B48: 4D 21 40 03 35 36 25 E5
$4B50: 6C 28 36 DF DB DB 93 27
$4B58: 24 D6 1F 20 2C 0D 6C 31
$4B60: 25 35 4D 3C 38 3F 3F 6F
$4B68: 28 6D 89 1C 1C FF 1B F7
$4B70: 1E DF 1B 24 24 2D 2D 2D
$4B78: 2D 2D 2D 2D 2D 2D 2D 36
$4B80: 36 36 36 36 36 36 36 36
$4B88: 36 3F 3F 3F 3F 3F 3F 3F
$4B90: 3F 3F 3F 24 24 24 24 24
$4B98: 24 24 24 00 DB DB DB 23
$4BA0: 24 24 24 24 2C 2D 2D 2D
$4BA8: 2D 2D 2D 2D 2D 2D 35 36
$4BB0: 36 36 36 36 36 36 36 36
$4BB8: 3E 3F 3F 3F 3F 3F 3F 3F
$4BC0: 3F 3F 27 24 24 24 24 4C
$4BC8: 49 49 C9 DB DB 23 24 24
$4BD0: 24 2C 2D 2D 2D 2D 2D 2D
$4BD8: 2D B5 30 36 36 36 36 36
$4BE0: 36 36 3F 3F 3F 3F 3F 3F
$4BE8: 3F 3F 24 24 24 6C 24 24
$4BF0: 24 2C 2D 2D 2D 2D 2D 35
$4BF8: 36 36 36 36 36 3E 3F 3F
$4C00: 3F 3F 3F 27 24 6C 24 24
$4C08: 24 2D 2D 2D 2D 36 36 36
$4C10: 36 3F 3F 3F 3F 64 21 24
$4C18: 2C 2D 35 36 3E 3F 0C 20
$4C20: 00 DB DB DB 23 24 24 24
$4C28: 24 2C 2D 2D 2D 2D 2D 2D
$4C30: 2D 2D 2D 35 36 36 36 36
$4C38: 36 36 36 36 36 3E 3F 3F
$4C40: 3F 3F 3F 3F 3F 3F 3F 27
$4C48: 24 24 24 24 4C 49 49 89
$4C50: 92 92 2A 1F 1C 3C 20 07
$4C58: 28 2D 2D 2D F5 F6 26 40
$4C60: 03 3F 3F 56 2D 0C 3F 3F
$4C68: 56 2D 0E DF 73 2D 04 40
$4C70: 18 40 18 23 24 24 FD 1C
$4C78: 1C 24 05 28 28 AD 15 36
$4C80: 1E C4 E3 16 0D 04 A8 C4
$4C88: FB 12 55 AA 92 2A 28 28
$4C90: E8 F3 0D 1E DF 3B 38 38
$4C98: 0E 05 38 38 04 00 DB DB
$4CA0: DB 2B 05 28 28 20 05 28
$4CA8: 05 28 A8 0E 15 2D 0E 0E
$4CB0: 65 05 28 0E 24 24 24 3C
$4CB8: 3F 3F 3F 3F 3F 3F 3F 3F
$4CC0: 3F 37 36 36 36 36 16 36
$4CC8: 36 36 36 2E 2D 2D 2D 2D
$4CD0: 2D 2D 2D 2D 2D 25 24 24
$4CD8: 24 24 24 FC DB 1A F7 F7
$4CE0: 17 BF FF 3F 27 2D 2D 05
$4CE8: 28 0C 65 05 28 0C 3F 17
$4CF0: 24 27 27 17 15 3F BF 29
$4CF8: 2D 35 3F 3F F7 2D 2D 2D
$4D00: DE 3F 3F F7 2D 2D 4D 89
$4D08: DA 2D 25 2F 28 25 2D 2C
$4D10: 25 25 35 3E 0E 96 92 D2
$4D18: DA 24 A8 44 22 07 F0 24
$4D20: 2D 25 BC 24 37 3E 24 96
$4D28: 12 18 24 3C 96 3A 24 2C
$4D30: 1F 36 36 6D 33 D5 6E 29
$4D38: 25 3F 05 28 24 FC DB DB
$4D40: DB 33 16 36 4D 23 24 04
$4D48: 68 3E 2E 3E 36 25 15 25
$4D50: 27 FD DB DB 21 27 24 D6
$4D58: 07 F8 B6 36 0D 24 2C 36
$4D60: 2E 24 15 26 40 18 40 18
$4D68: 40 18 40 18 D8 03 40 18
$4D70: 2B 15 6D 49 49 2D 0C 75
$4D78: 04 00 D8 D8 D8 D8 D8 48
$4D80: D8 48 D8 C8 2D 2D 2D 2D
$4D88: 2D 2D 2D 2D 2D 2D 3E 1B
$4D90: 1B 1B 1B 1B 1B 1B 1B 1B
$4D98: 33 0D 09 09 2D 0D 09 2D
$4DA0: 0D 09 09 3E 1B 3B 1F 1B
$4DA8: 3F 1F 1B 3F 1B 33 0D 29
$4DB0: 09 09 29 29 09 09 29 09
$4DB8: 3E 1B 1F 1B 1B 1F 1F 1B
$4DC0: 1B 1F 33 0D 0D 09 09 0D
$4DC8: 09 0D 09 09 09 0D 3E 3B 1B
$4DD0: 1B 3B 1B 3B 1B 1B 3B 33
$4DD8: 0D 0D 09 2D 0D 09 2D 0D
$4DE0: 09 0D 3E 1B 3F 1F 1B 3F
$4DE8: 1F 1B 3F 1F 33 0D 29 09
$4DF0: 09 29 29 09 09 29 09 3E
$4DF8: 1B 3F 1F 1B 3F 1F 1B 3F
$4E00: 1F 33 0D 0D 09 2D 0D 09
$4E08: 2D 0D 09 0D 3E 3B 1B 1B
$4E10: 3B 1B 3B 1B 1B 3B 33 0D
$4E18: 0D 09 09 0D 09 0D 09 09
$4E20: 0D 3E 1B 1F 1B 1B 1F 1F
$4E28: 1B 1B 1F 33 0D 29 09 09
$4E30: 29 29 09 09 29 09 3E 1B
$4E38: 3B 1F 1B 3F 1F 1B 3F 1B
$4E40: 33 0D 09 09 2D 0D 09 2D
$4E48: 0D 09 09 3E 1B 1B 1B 1B
$4E50: 1B 1B 1B 1B 1B 33 2D 2D
$4E58: 2D 2D 2D 2D 2D 2D 2D 2D
$4E60: 06 00 92 92 12 2D 25 3F
$4E68: 3F 3E 3C 26 08 2D 2D 5C
$4E70: 3F 3F 0C 2D 3C 27 2D 3C
$4E78: 3F 28 2D 25 3F 3F 2C 2D
$4E80: E5 3F DF DB DB 24 24 24
$4E88: 2C 2D 2D 2D 2D 2D 2D 2D
$4E90: 2D 2D 35 36 36 36 36 36
$4E98: 36 36 36 36 3E 3F 3F 3F
$4EA0: 3F 3F 3F 3F 3F 3F 27 24
$4EA8: 24 24 24 24 6C 49 49 92
$4EB0: 12 68 09 04 00 DB DB 1B
$4EB8: 24 35 66 3C 68 12 2D 2D
$4EC0: 2D 25 3F 0D 08 2D 2E F5
$4EC8: 3F 27 B5 35 37 18 2E 2D
$4ED0: 2D 3C 6F 09 24 24 24 24
$4ED8: 24 3C 3F 3F 3F 3F 3F 3F
$4EE0: 3F 3F 3F 37 36 36 36 36
$4EE8: 36 36 36 36 36 2E 2D 2D
$4EF0: 2D 2D 2D 2D 2D 2D 2D 25
$4EF8: 24 24 24 BC D2 F3 9B DB
$4F00: DF 1B 40 18 B0 DA 1B 04
$4F08: 40 18 40 18 40 18 40 18
$4F10: 4D 49 11 4D 41 21 00 DB
$4F18: DB DB 23 24 24 24 24 2C
$4F20: 2D 2D 2D 2D 2D 2D 2D 2D
$4F28: 2D 35 36 36 36 36 36 36
$4F30: 36 36 36 3E 3F 3F 3F 3F
$4F38: 3F 3F 3F 3F 3F 27 24 24
$4F40: 24 24 45 24 05 20 05 28
$4F48: 28 28 05 28 2D 75 75 0E
$4F50: 0E 76 36 36 17 BE 17 17
$4F58: F7 3F 3F 1C E7 1C 1C 3C
$4F60: 20 4C 49 49 1C 1C 24 2C
$4F68: 28 B0 92 92 A9 0E 36 17
$4F70: 17 DD 3F DC 2D 2D 65 3F
$4F78: 3F 3F E7 2D 2D 2D 2D 3C
$4F80: 3F 3F 3F E7 2D 2D 2D 2D
$4F88: 1C 3F 3F 3F 27 2D 2D 2D
$4F90: 1C 3F 3F 27 2D 2D 3C 3F
$4F98: 2C 2D 3C 67 2D 2C 04 00
$4FA0: D8 D8 D8 D8 D8 48 D8 48
$4FA8: D8 C8 2D 2D 2D 2D 2D 2D
$4FB0: 2D 2D 2D 2D 3E 1B 1B 1B
$4FB8: 1B 1B 1B 1B 1B 1B 33 0D
$4FC0: 09 09 09 09 09 09 09 09
$4FC8: 09 3E 1B 1B 1B 1B 1B 1B
$4FD0: 1B 1B 1B 33 0D 09 29 2D
$4FD8: 2D 2D 2D 2D 09 09 3E 1B
$4FE0: 3B 3F 3F 3F 3F 3F 3F 1B
$4FE8: 33 0D 29 2D 0D 29 2D 09
$4FF0: 2D 2D 09 3E 1B 3F 1F 3B
$4FF8: 3F 3F 1B 3F 1F 33 0D 29
$5000: 2D 29 0D 09 2D 29 2D 09
$5008: 3E 1B 1B 3B 1F 1F 1F 3F
$5010: 1B 1B 33 0D 09 29 0D 0D
$5018: 09 0D 2D 09 09 3E 1B 1B
$5020: 3F 1B 3B 1B 3B 1F 1B 33
$5028: 0D 09 29 0D 0D 09 0D 2D
$5030: 09 2D 3E 3B 1F 3F 1F 1F
$5038: 1F 3F 1F 1B 33 0D 09 29
$5040: 2D 0D 09 2D 2D 2D 09 3E
$5048: 1B 1B 3F 3F 3F 3F 3F 1F
$5050: 1B 33 0D 09 09 09 09 09
$5058: 09 09 09 09 3E 1B 1B 1B
$5060: 1B 1B 1B 1B 1B 1B 33 0D
$5068: 09 09 09 09 09 09 09 09
$5070: 09 3E 1B 1B 1B 1B 1B 1B
$5078: 1B 1B 1B 33 2D 2D 2D 2D
$5080: 2D 2D 2D 2D 2D 2D 06 00
$5088: DB DB DB 23 24 24 24 24
$5090: 2C 2D 2D 2D 2D 2D 2D 2D
$5098: 2D 2D 35 36 36 36 36 36
$50A0: 36 36 36 36 3E 3F 3F 3F
$50A8: 3F 3F 3F 3F 3F 3F 27 24
$50B0: 24 24 24 8D 92 22 2C 28
$50B8: 28 75 0E 36 1E 1E 3F 1C
$50C0: 4D 49 49 49 49 F3 3F 1C
$50C8: 07 20 64 05 28 75 15 36
$50D0: 04 40 18 40 18 40 18 40
$50D8: 18 32 BE 1E 3F 1C 1C 24
$50E0: 05 28 28 AD DF DB DB DB
$50E8: 03 28 75 15 36 17 17 3F
$50F0: 1C 1C 24 05 68 92 92 09
$50F8: 36 15 15 2D 05 28 20 3C
$5100: 38 38 F7 04 40 18 01 28
$5108: 2D 28 F8 DB AB 4D 49 49
$5110: A8 15 96 F2 1E 36 15 15
$5118: 96 BA 17 DF E3 1C 3F 1E
$5120: 1E DF E3 1C 04 40 18 0C
$5128: 05 20 3C 38 20 40 18 05
$5130: 28 20 00 DB DB DB 23 24
$5138: 24 24 24 2C 2D 2D 2D 2D
$5140: 2D 2D 2D 2D 2D 35 36 36
$5148: 36 36 36 36 36 36 36 3E
$5150: 3F 3F 3F 3F 3F 3F 3F 3F
$5158: 3F 27 24 24 24 24 4C 05
$5160: 30 2E 24 35 76 20 2C 36
$5168: 25 2C 36 25 2C 36 25 2C
$5170: 36 DF DB 2A F5 2D 1E 2D
$5178: 1E 2D 1E 2D 04 40 18 40
$5180: 18 D8 DB 2B E5 2D 1C 2D
$5188: 07 28 E5 2D 96 92 21 05
$5190: 28 30 36 FD B4 12 05 28
$5198: 3E 0D 1E B5 DA 40 18 40
$51A0: 18 40 18 40 18 18 35 25
$51A8: 2C 1F DF DB DB DB 1B 05
$51B0: 28 75 35 BF 27 27 2D B6
$51B8: 92 92 92 92 02 38 27 27
$51C0: 2C 35 2D 3E E7 04 00 40
$51C8: 18 40 18 2D 0E 2D 28 15
$51D0: F6 1E DF DB 1B 40 18 09
$51D8: 38 1E F7 40 03 F7 76 15
$51E0: 4D 28 0E 05 28 15 F6 E7
$51E8: 17 E7 B4 49 FE 4D F1 3F
$51F0: 17 76 2D 05 20 FC 1B 04
$51F8: C0 1B 15 AE 6E 49 21 05
$5200: 20 05 F0 13 2C 20 2C 05
$5208: 20 F7 17 3C 3C BE 24 37
$5210: 27 1E BE 76 26 40 18 3B
$5218: 1C 2C 15 DF DB 24 24 24
$5220: 2D 2D 2D 2D 2D 2D 2D 2D
$5228: 2D 2D 36 36 36 36 36 36
$5230: 36 36 36 36 3F 3F 3F 3F
$5238: 3F 3F 3F 3F 3F 3F 24 24
$5240: 24 24 24 24 24 00 DB DB
$5248: DB 23 24 24 24 24 2C 2D
$5250: 2D 2D 2D 2D 2D 2D 2D 2D
$5258: 35 36 36 36 36 36 36 36
$5260: 36 36 3E 3F 3F 3F 3F 3F
$5268: 3F 3F 3F 3F 27 24 24 24
$5270: 24 4C 49 49 49 01 40 18
$5278: 40 18 08 2D 0E 0E 36 17
$5280: 17 3F 1C 1C 24 05 F8 01
$5288: 38 38 BF 17 36 0E 15 2D
$5290: 3C E7 24 05 28 75 3F 37
$5298: 2E 2C B6 52 49 F1 1E 1E
$52A0: 1E 1E 1E 1E 1E 24 24 24
$52A8: 24 24 37 BE 6C 89 92 1A
$52B0: 0D 18 1C 04 A8 15 05 38
$52B8: 38 38 68 15 15 05 38 20
$52C0: 00 DB DB DB 23 24 24 24
$52C8: 24 2C 2D 2D 2D 2D 2D 2D
$52D0: 2D 2D 2D 35 36 36 36 36
$52D8: 36 36 36 36 36 3E 3F 3F
$52E0: 3F 3F 3F 3F 3F 3F 3F 27
$52E8: 24 24 24 24 95 92 52 2D
$52F0: DC 2C 28 28 28 28 28 28
$52F8: 28 28 28 28 28 28 28 28
$5300: 35 BE 17 17 17 1E 1E 1E
$5308: 1E 1E 1E 1E 1E 1E 3F 2C
$5310: 3D 28 AC 24 15 24 15 24
$5318: 15 24 15 24 15 24 15 24
$5320: 0E 24 15 24 0E 24 0E 24
$5328: 15 B4 92 92 92 92 0A 3F
$5330: 1C 07 38 38 38 20 25 15
$5338: 15 35 15 15 15 3C 38 38
$5340: 18 38 B0 77 35 35 35 25
$5348: 40 18 40 18 DB DB DB 1B
$5350: 40 18 40 18 C0 DB 2D D7
$5358: AE 15 15 15 15 65 3C 38
$5360: 38 38 38 37 35 35 35 35
$5368: 25 07 38 38 20 00 92 92
$5370: 92 3A 3F 3F 3F 3F 27 24
$5378: 24 24 24 24 24 24 24 24
$5380: 2C 2D 2D 2D 2D 2D 2D 2D
$5388: 2D 2D 35 36 36 36 36 36
$5390: 36 36 36 36 3E 3F 3F 3F
$5398: 3F 47 09 25 27 25 27 25
$53A0: 27 25 27 25 27 25 27 25
$53A8: 27 25 27 25 27 FD DB DB
$53B0: 5B 29 3E 2E 3E 2E 3E 2E
$53B8: 3E 2E 3E 2E 3E 2E 3E 2E
$53C0: 3E 2E 3E 2E 1F 40 18 C0
$53C8: DB 2C 2E 2C 2E 6C 31 25
$53D0: 35 25 35 4D 2C 2E 2C 2E
$53D8: 24 40 18 40 18 02 80 3E
$53E0: 3C 3E 3C FE 23 37 27 37
$53E8: 27 DF 3E 3C 3E 3C 26 00
$53F0: DB DB DB 23 24 24 24 24
$53F8: 2C 2D 2D 2D 2D 2D 2D 2D
$5400: 2D 2D 35 36 36 36 36 36
$5408: 36 36 36 36 3E 3F 3F 3F
$5410: 3F 3F 3F 3F 3F 3F 27 24
$5418: 24 24 24 4C 24 05 20 05
$5420: 28 28 28 05 28 2D 75 75
$5428: 15 15 76 36 36 1E F6 1E
$5430: 1E BF 0B 3F 3F 1C E7 1C
$5438: 1C 3C 20 4D 49 09 28 28
$5440: 15 3F 1E 0E 15 2E 2C E8
$5448: 5B BC 3C DF C3 AF 39 F7
$5450: 30 0E AE 54 07 28 18 05
$5458: 38 27 18 38 68 E0 25 2D
$5460: 38 4D 89 3A 2E 05 20 1C
$5468: CD 35 0E 3E 05 40 03 AC
$5470: 96 13 35 0E 40 03 3C 2C
$5478: 3C 8E 21 00 DB DB DB 23
$5480: 24 24 24 24 2C 2D 2D 2D
$5488: 2D 2D 2D 2D 2D 2D 35 36
$5490: 36 36 36 36 36 36 36 36
$5498: 3E 3F 3F 3F 3F 3F 3F 3F
$54A0: 3F 3F 27 24 24 24 24 4D
$54A8: 49 49 C9 40 18 40 18 40
$54B0: 18 FD 1E 2D 2D 0E 3F 3F
$54B8: 3F 2E 2D 2D 35 3F 3F 3F
$54C0: 0E 2D 2D 1E 3F 0E 36 2D
$54C8: 2C 30 2E 24 24 2C 35 37 2D
$54D0: AC 3E 3F 2E 2D 1E 3F 0E
$54D8: 3F 24 37 07 F8 1B 08 27
$54E0: 16 1E 24 24 07 38 F7 2D
$54E8: 35 3F 3F 2E 2D 35 3F 3F
$54F0: 0E 2D 35 3F 4D 49 40 03
$54F8: 36 36 6F 2E 2E DF 38 3E
$5500: F4 4D 21 00 1B 25 E5 2B
$5508: 2D 2D 2D 2D FD DB 13 2D
$5510: 04 38 B7 12 3E 3C 3E 3C
$5518: 3E 3C 3E 24 24 2C 1F 3F
$5520: 3F 4D 91 12 AF DF 23 24
$5528: 24 14 24 24 24 2C 2D 2D
$5530: 2D 2D 2D 2D 2D 2D 2D 35
$5538: 36 36 36 36 36 36 36 36
$5540: 36 3E 3F 3F 3F 3F 3F 3F
$5548: 3F 3F 3F 27 24 24 24 8D
$5550: 6A 91 4D 49 80 4D 49 09
$5558: 38 1E 17 17 2E 2C 28 28
$5560: 16 F7 04 40 18 40 18 04
$5568: 40 18 40 18 C0 DF DB DB
$5570: 9B 04 00 DB DB DB 23 24
$5578: 24 24 24 2C 2D 2D 2D 2D
$5580: 2D 2D 2D 2D 2D 35 36 36
$5588: 36 36 36 36 36 36 36 3E
$5590: 3F 3F 3F 3F 3F 3F 3F 3F
$5598: 3F 27 24 24 24 24 0D 48
$55A0: 49 49 24 24 24 B4 92 52
$55A8: 1E 36 36 36 04 40 18 40
$55B0: 18 09 2D 2D FD DB DB 3F
$55B8: 3F 3F 4D 49 38 38 38 38
$55C0: 38 48 49 49 49 C9 DA D2
$55C8: D3 2A 28 28 28 B0 92 92
$55D0: 12 07 38 38 38 F8 1E 1E
$55D8: 1E 1E 04 40 18 48 08 E7
$55E0: E7 4D 89 3C 20 07 68 49
$55E8: F1 BE AE 2D 28 05 B0 D2
$55F0: DB AD 75 DE DB 40 03 76
$55F8: 76 DF DB 05 20 05 20 07
$5600: 38 1E 1E B8 04 00 2D E5
$5608: 3F 2C 1C 1E 36 27 17 04
$5610: 00
```

Verkettung von Applesoft-Programmen

Variablenübergabe unter DOS 3.3

von Frank Bühler

Eine der Methoden, mehrere Programme hintereinander ablaufen zu lassen, besteht darin, eine Liste von RUN-Anweisungen in einer sog. EXEC-Datei abzuspeichern. Ein gravierender Nachteil dieses Verfahrens ist jedoch, daß die Inhalte von einfachen Variablen und Variablenfeldern nicht übergeben werden können. Außerdem sind ohne Kunstgriffe keine Tastatureingaben vom Benutzer während des Programmlaufes möglich. Wie man eine Verkettung von Applesoft-Programmen ohne Verlust der Variablenwerte und ohne eine gesonderte EXEC-Datei mit einem kleinen Trick erreichen kann, beschreibt dieser Artikel.

1. Grundgedanke

Um das obengenannte Problem zu lösen, ist die Kenntnis, wie die Variablen intern im Programmspeicher abgelegt sind, von besonderer Bedeutung.
Bekanntlich werden mit Hilfe von LOMEM: ($0069, $006A) und HIMEM: ($0073, $0074) die niedrigste bzw. höchste Adresse des Speicherbereiches festgelegt, der einem Applesoft-Programm zur Verfügung steht (einschließlich der Variablen). Einfache Variablen werden von LOMEM: an aufwärts und Zeichenfolgen von HIMEM: an abwärts gespeichert. Daneben sind noch weitere Pointer von Bedeutung, die den Beginn und das Ende der Feldvariablen, das Ende des String-Bereichs usw. festhalten. Diese liegen alle im Bereich von $0067 bis $0074 (vgl. **Tabelle 1**). Bei Eingabe von CLEAR, RUN, RUN <Programmname> u.a. werden diese Pointer zurückgesetzt, wobei die im Programmspeicher befindlichen Variablenwerte unverändert vorliegen, also nicht gelöscht werden. Der Lösungsansatz besteht darin, die Pointer vor dem Zurücksetzen zu retten und anschließend das nachfolgende Programm zu starten, welches die Pointer auf die ursprünglichen Werte setzt. Damit stehen dem aufgerufenen Programm wieder die alten Variablenwerte zur Verfügung.

2. Realisierung und Anwendung

2.1. Erweiterter CALL-Befehl

Um eine größtmögliche Flexibilität und einen gewissen Komfort zu erreichen, wurde Gebrauch von der erweiterten CALL-Anweisung gemacht, die hier folgendes allgemeine Format hat:
CALL Adresse, String-Name
Im Gegensatz zu Integer-BASIC erfolgt in Applesoft die Überprüfung auf Richtigkeit der Syntax erst bei Ausführung dieser Anweisung. Mit Hilfe bereits implementierter Interpreter-Routinen können die nachfolgenden Zeichen eingelesen und der Textpointer auf den nächsten auszuführenden Befehl gesetzt werden, um so die Meldung „SYNTAX ERROR" zu vermeiden. Beispielsweise ist es mit Hilfe der CHKCOM-Routine (CHecKCOMma) möglich, das aktuelle Applesoft-Zeichen bezüglich eines Kommas zu prüfen und bei Übereinstimmung den Textpointer zu erhöhen. Andernfalls übernimmt die Routine auch die Ausgabe einer Fehlermeldung (SYNTAX ERROR).

2.2. Finden einer Zeichenkette

In Applesoft benötigt jede einfache String-Variable 7 Bytes (bei Feldvariablen 3 Bytes pro Eintragung) zur Festlegung des Namens (2 Bytes), der Länge (1 Byte) und der Adresse (2 Bytes), ab der die Zeichenkette abgelegt ist (die restlichen 2 Bytes sind Null). Dazu kommt pro zugewiesenem Zeichen 1 Byte, das mit Bit 7 off abgespeichert ist (A$ = "Ap" + "ple" benötigt also insgesamt 13 Bytes).
Wie findet nun das Maschinenprogramm **RUN.FILE** die Zeichenkette des Programmnamens? Dies geschieht mittels der bereits implementierten ROM-Routine PTRGET (PoinTeRGET), nach deren Aufruf in $0083 (Low-Byte) und $0084 (High-Byte) die Adresse auf den Variablen-Eintrag abgelegt ist. Dort steht dann die Länge und Startadresse des gespeicherten Strings. Indirekt indiziert mit dem Y-Register können die wichtigen Informationen in die freien Zero-Page-Speicherstellen $00FA (Länge), $00FB (Low-Byte) und $00FC (High-Byte der Adresse der Zeichenkette) zwischengespeichert werden.

2.3. Anwendung

Wie verwendet man nun diese Routine? Zunächst müssen LOMEM: und HIMEM: auf genügend große Werte gesetzt werden (z.B. für LOMEM: 16384, hinter HGR-Bild), da normalerweise die einfachen Variablen unmittelbar hinter dem Applesoft-Programm abgelegt werden und so durch das Laden eines längeren Programmes nicht überschrieben werden können.
Nachdem das Maschinenprogramm mit „BLOAD RUN.FILE" geladen wurde, kann mit CALL „780,A$" das nächste Programm gestartet werden, wobei in A$ der Programmname des aufgerufenen Programmes abgelegt sein muß (s. Startprogramm **START**).
Die erste ausführbare Anweisung im aufgerufenen Programm muß nun „CALL 837" lauten. Dadurch werden die geretteten Pointer wieder hergestellt und stehen diesem Programm zur Verfügung (s. Zeile 120 im Hauptprogramm **HP**).

Mit Hilfe dieser Routine ist es aber auch möglich, „echte" Unterprogramme zu schreiben. Die Vorgehensweise ist dieselbe wie oben beschrieben, nur daß nach Ende des aufgerufenen Programms das rufende Programm wieder gestartet wird. Abhängig von bestimmten Variablenwerten oder durch Verwendung von berechneten Sprungbefehlen (s. GOTO-Routine: „CALL 768") kann unmittelbar hinter die Zeile gesprungen werden, in der der Programmaufruf steht. Die Programme START, HP, UP1 und UP2 zeigen die Vor-

gehensweise. Damit ist eine vertikale und horizontale Verkettung von Applesoft-Programmen möglich.
Der größte Nutzen dieser Routine besteht darin, mehrere kleine Programme schreiben zu können, die überschaubarer und damit schneller realisiert werden können. Beispielsweise ist die Aufsplittung eines größeren Programmes in drei kleinere Programme für die Eingabe, Verarbeitung und Ausgabe sinnvoll. Außerdem kann, falls z.B. ein Programm um eine Druckroutine erweitert werden soll, ein gesondertes Programm geschrieben werden, ohne große Änderungen in dem ursprünglichen Programm vornehmen zu müssen.

Damit auch Zeichenketten, die im Programmtext liegen (z.B. A$ = "Apple"), richtig übergeben werden können, müssen diese von HIMEM: an abwärts gespeichert sein (die String-Adresse würde sonst auf die entsprechende Stelle im Programmtext verweisen). Durch einen kleinen Trick kann man dies immer gewährleisten, indem man bei einer Wertzuweisung zusätzlich + "" schreibt (s. Zeile 260 im Hauptprogramm). Wer will, kann einmal die Programme mit Zeile 260 und einmal ohne diese Zeile starten.

Anm.d.Red.: Die beschriebenen Routinen funktionieren nur unter DOS 3.3. Unter dem ProDOS-BASIC.SYSTEM sind sie ohnehin entbehrlich, weil ein funktionsfähiger CHAIN-Befehl implementiert worden ist. Für DOS 3.3 gab es zwar auch einen CHAIN-Befehl als auf der System-Master-Disk von 1980/1982 enthaltenes Maschinenprogramm, doch funktionierte dies bei Strings nicht immer korrekt. us

Kurzhinweise

1. Zweck:
Vertikale und horizontale Verkettung von Applesoft-Programmen ohne Verlust der Variablen
2. Konfiguration:
II+, IIe oder IIc;
DOS 3.3; nicht ProDOS-BASIC.SYSTEM wegen HIMEM-Änderung!
3. Test
RUN START
4. Sammeldisk:
START
HP
UP1
UP2
(Applesoft-Demo-Programme)
T.RUN.FILE
(Big-Mac-Quelltext)
RUN.FILE
(Maschinenprogramm)

Tabelle 1

Adresse	Bedeutung
$0067 (103):	Anfang des Programmtextes
$0069 (105):	LOMEM:
$006B (107):	Anfang der Feldvariablen
$006D (109):	Ende der Feldvariablen
$006F (111):	Anfang des String-Bereichs
$0071 (113):	Zeiger bei String-Verwaltung
$0073 (115):	HIMEM:

(Die Pointer umfassen jeweils zwei Bytes. Die Zahlen in Klammern sind die entsprechenden Dezimalwerte.)

START

```
100 REM START
110 TEXT : HOME
120 PRINT "S T A R T P R O G R A M M"
130 LOMEM: 16384: HIMEM: 24576
140 CLEAR
150 D$ = CHR$ (13) + CHR$ (4):
    REM Ctrl-M, Ctrl-D
160 GOTO = 768:START = 780
170 PRINT D$"BLOAD RUN.FILE"
180 FLAG = 0:ZAEHLER = 0
190 FILE$ = "HP"
200 CALL START,FILE$
```

HP

```
100 REM Hauptprogramm-Demo
110 REM Pointer restaurieren
120 CALL 837
130 TEXT : HOME
140 PRINT "H A U P T P R O G R A M M"
150 :
160 REM Nicht beim 1. Durchlauf
170 IF FLAG = 1 THEN 300
180 IF FLAG = 2 THEN CALL GOTO,WERT
190 :
200 REM ********************
210 FLAG = 1
220 :
230 PRINT "Wertzuweisung"
240 AA = 1111:AA(1,1,1) = 123
250 AA$ = "Apple-Computer":AA$(1)
    = "Software"
260 AA$ = AA$ + "":AA$(1) = AA$(1) + ""
270 :
280 REM ********************
290 :
300 ZAEHLER = ZAEHLER + 1
310 VTAB 4: HTAB 1: PRINT "Zähler: ";
    ZAEHLER
320 IF ZAEHLER > 5 THEN 440: REM UP2
330 :
340 VTAB 8: HTAB 1
350 PRINT AA$;: HTAB 20: PRINT AA$(1)
360 VTAB 10: HTAB 1
370 PRINT "AA: ";AA;: HTAB 20:
    PRINT "AA (1,1,1): ";AA(1,1,1)
380 :
390 GET WEITER$: PRINT
400 REM Unterprogramm 1 laden
410 FILE$ = "UP1"
420 CALL START,FILE$
430 :
440 GET WEITER$: PRINT
450 REM Unterprogramm 2 laden
460 FILE$ = "UP2"
470 FLAG = 2:WERT = 500
480 CALL START,FILE$
490 :
500 VTAB 14: HTAB 1: PRINT
    "Weiter nach Rückkehr von UP2"
510 PRINT AA$;AA$(1)
520 PRINT : PRINT "ENDE DEMO": END
```

UP1

```
100 REM UP1
110 REM Pointer restaurieren
120 CALL 837
130 TEXT : HOME
140 PRINT "U N T E R P R O G R A M M  1"
150 :
160 VTAB 4: HTAB 1
170 PRINT "Werte:": PRINT "-----"
180 VTAB 8: HTAB 1
190 AA = AA + 1:AA(1,1,1) = AA(1,1,1)
    * 2
200 PRINT "AA + 1 = ";AA;: HTAB 20:
    PRINT "AA (1,1,1) * 2 = ";
    AA(1,1,1)
210 PRINT AA$;: HTAB 20: PRINT AA$(1)
220 :
230 REM Warteschleife
240 FOR DUMMY = 1 TO 2000: NEXT
250 :
260 FILE$ = "HP"
270 REM RETURN
280 CALL START,FILE$
```

UP2

```
100 REM UP2
110 REM Pointer restaurieren
120 CALL 837
130 TEXT : HOME
140 PRINT "U N T E R P R O G R A M M  2"
150 :
160 VTAB 4: HTAB 1
170 PRINT "Werte:": PRINT "-----"
180 VTAB 8: HTAB 1
190 AA = AA + 1:AA(1,1,1) = AA(1,1,1)
    * 2
200 PRINT "AA + 1 = ";AA;: HTAB 20:
    PRINT "AA (1,1,1) * 2 = ";
    AA(1,1,1)
210 AA$ = "AA$/AA$(1) neu belegt":AA$(1)
    = " in UP2"
220 AA$ = AA$ + "":AA$(1) = AA$(1) + ""
230 PRINT AA$;AA$(1)
240 :
250 REM Warteschleife
260 FOR DUMMY = 1 TO 2000: NEXT
270 :
280 FILE$ = "HP"
290 REM RETURN
300 CALL START,FILE$
```

Die Variable GOTO (geschrieben mit 0) ist nicht mit dem Befehl GOTO (geschrieben mit „O") zu verwechseln.

RUN.FILE

BSAVE RUN.FILE, A$0300, L$0055

```
1   ****************************
2   *                          *
3   * RUN mit Variablenübergabe *
4   *                          *
5   ****************************
6   *
7   * Definitionen:
8   * -------------
9   *
10  TXTTAB   EQU  $67
11  VARPNT   EQU  $83
12  SAVE     EQU  $FA
13  COUT     EQU  $FDED
14  CHKCOM   EQU  $DEBE
15  FRMEVL   EQU  $DD7B
16  GETADR   EQU  $E752
17  GOTO     EQU  $D941
18  PTRGET   EQU  $DFE3
19  *
20           ORG  $300
21  *
22  * GOTO-Routine:
23  * -------------
```

```
                24   * (berechneter
                25   * Sprungbefehl)
                26   *
                27   * Prüfung auf Komma
                28   *
Ø3ØØ: 2Ø BE DE  29   GØTØ     JSR  CHKCOM
                3Ø   *
                31   * Ausdruck berechnen
                32   *
Ø3Ø3: 2Ø 7B DD  33            JSR  FRMEVL
Ø3Ø6: 2Ø 52 E7  34            JSR  GETADR
                35   *
                36   * GOTO-Routine aufrufen
                37   *
Ø3Ø9: 4C 41 D9  38            JMP  GOTO
                39   *
                4Ø   * Hauptroutine:
                41   * ------------
                42   *
                43   * 1. Var.-Pointer retten
                44   * 2. String-Name suchen
                45   * 3. Programm starten
                46   * 4. Pointer restaur.
                47   *
                49   * Pointer retten
                5Ø   *
Ø3ØC: A2 ØD     51   START    LDX  #13
Ø3ØE: B5 67     52   WDH1     LDA  TXTTAB,X
Ø31Ø: 9D 55 Ø3  53            STA  POINTER,X
Ø313: CA        54            DEX
Ø314: 1Ø F8     55            BPL  WDH1
                56   *
                57   * Prüfung auf Komma
                58   *
Ø316: 2Ø BE DE  59            JSR  CHKCOM
                6Ø   *
                61   * String-Suche
                62   * Adresse u. Länge retten
                63   *
Ø319: 2Ø E3 DF  64            JSR  PTRGET
Ø31C: AØ Ø2     65            LDY  #$Ø2
Ø31E: B1 83     66   WDH2     LDA  (VARPNT),Y
Ø32Ø: 99 FA ØØ  67            STA  SAVE,Y
Ø323: 88        68            DEY
Ø324: 1Ø F8     69            BPL  WDH2
                7Ø   *
                71   * Programm starten
                72   *
                73   * RUN-Befehl
                74   *
Ø326: AØ Ø4     75            LDY  #$Ø4
Ø328: B9 5Ø Ø3  76   WDH3     LDA  RUNCMD,Y
Ø32B: 2Ø ED FD  77            JSR  COUT
Ø32E: 88        78            DEY
Ø32F: 1Ø F7     79            BPL  WDH3
                8Ø   *
                81   * Programmname
                82   *
Ø331: A6 FA     83            LDX  SAVE
Ø333: AØ ØØ     84            LDY  #$ØØ
Ø335: B1 FB     85   WDH4     LDA  (SAVE+1),Y
Ø337: Ø9 8Ø     86            ORA  #%1ØØØØØØØ
Ø339: 2Ø ED FD  87            JSR  COUT
Ø33C: C8        88            INY
Ø33D: CA        89            DEX
Ø33E: DØ F5     9Ø            BNE  WDH4
                91   *
                92   * Carriage Return
                93   *
Ø34Ø: A9 8D     94            LDA  #$8D
Ø342: 4C ED FD  95            JMP  COUT
                96   *
                97   * Pointer herstellen
                98   *
Ø345: A2 ØD     99            LDX  #13
Ø347: BD 55 Ø3  1ØØ  WDH5     LDA  POINTER,X
Ø34A: 95 67     1Ø1           STA  TXTTAB,X
Ø34C: CA        1Ø2           DEX
Ø34D: 1Ø F8     1Ø3           BPL  WDH5
Ø34F: 6Ø        1Ø4           RTS
                1Ø5  *
                1Ø6  * "RUN" zum Starten von
                1Ø7  * A-Files
                1Ø8  * rückwärts geschrieben
Ø35Ø: CE D5 D2  1Ø9  RUNCMD   ASC  "NUR"
                11Ø  *
                111  * Ctrl-M, Ctrl-D
                112  *
Ø353: 84 8D     113           DFB  $84,$8D
                114  *
                115  POINTER  EQU  *
```


Software Preissenkung

Wir stellen die Preise auf den Kopf

Merlin Pro Macro Assembler ..... ~~448,-~~ **199,-**
Merlin ..... ~~250,-~~ **150,-**
Merlin Combo **250,-**
Mousewrite .. ~~498,-~~ **349,-** – der Apple *IIe*® wird zum Mac®

Chart'n Graph Toolbox 110,- **Database Toolbox** **110,-**
Video Toolbox ..... **110,-** **Wizard's Toolbox** **110,-**
Munch A Bug ..... **130,-** **Printographer** **130,-**

ZUSATZ-KARTEN:

V-24-Schnittstelle ..... **199,-** **Z-80-Karte** **98,-**
80-Zeichen-Karte m.Softswitch **236,-** **16 K-Language-Karte** **98,-**
Joy Stick De Luxe ..... **59,-** **Accelerator** 3,6 MHz **950,-**
68000 Intemex ..... **1600,-** **PAL Karte** **110,-**
RGB Karte ..... **239,-** **IEEE 488** **312,-**
Koppler dataphon m. FTZ ..... **325,-** **Z 80 B Karte** mit Software **919,-**
Centronics-Karte von Epson für Graphik .... **210,-** für Text **145,-**
Centronics-Schnittstelle für 2 Drucker gleichzeitig **129,-**

Super-Eprommer

belegt keinen Slot, incl. Software für 2716-27128 ..... **239,-**

Floppy-Controller

FDC 4 für alle Laufwerke ..... **169,-** Bausatz wie links **159,-**
Leerplatine wie oben incl. Prom u. Eprom **98,-**

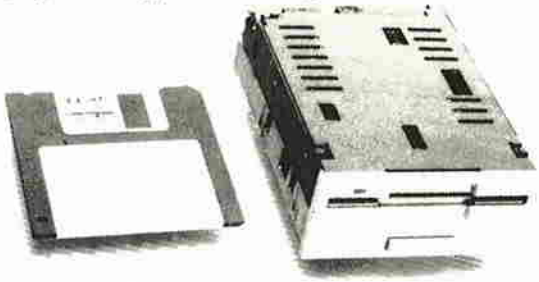
Erphi-Controller **298,-**
Disketten 1D, 48 tpi ..... 10 St. **29,-** **Disketten** 2D, 48 tpi ..... 10 St. **36,-**

Frei programmierbare Keyboards

Wir bieten Ihnen die **Preh-Qualität** auch für Apple. AK 87 spez. mit Gehäuse, Anschlußkabel, Zehner-Tastenfeld, dt. Zeichensatz, Sondertasten für Ctrl-Codes und Rechenfunktionen **339,-**
Preh Commander Keyboard, frei programmierbar bis zu 10 Ebenen, pro Taste bis zu 250 Zeichen ..... nur **599,-**
Gleiche Tastatur wie oben **für Apple IIe** ..... nur **698,-**

TEAC 3½" Laufwerk FD 35 F 498,-

Speicherkapazität 1 MB, (formatiert 640 KB) jetzt für nur .....



TEAC FD 55 AV 1 x 40 Track .... **395,-** **TEAC FD 55 BV** 2 x 40 Track **460,-**
TEAC FD 55 EV 1 x 80 Track .... **445,-** **TEAC FD 55 FV** 2 x 80 Track **398,-**

Panasonic Drucker: **1592** ..... nur **1599,-**
1091 ..... nur **1095,-** **1092** ..... nur **1195,-**

Die Microfloppy mit Zukunft:

Speicherkapazität: 2 x 1 MByte formatiert: 2 x 640 kByte. Anschlußfertig mit PROM-residenter Patchsoftware für CP/M 2.2, Apple DOS 3.3, DiversiDOS 2-C, 4-C (DD MOVER), Apple Pascal 1.1, Pascal 1.2, Pro-DOS 1.0.1, 1.1, 1.1.1 zum Preis von **1498,-**
Low Power Version **1598,-**

Gesamt-Preisliste anfordern!
Preise inclusive gesetzlicher Mehrwertsteuer.
Händlerpreisliste bitte schriftlich anfordern!

UEDING electronics

Holtewiese 2
5750 Menden 1

DFÜ 02373/66877
Tel. 02373/63159

Band 1:
1. Einleitung
2. Was Sie benötigen
3. Starten von APPLE WORKS
4. Der Schreibtischmanager
5. Datenbank
6. Rechenblatt

A1 Anschluß der Festplatte ProFile
A2 APPLE II Easy Pieces Referenz
A3 Druckeranpassungen
A4 DOS 3.3 Konvertierungen
A5 APPLE WORKS Disketten sichern/kopieren
A6 Hilfsfunktionen nach Programmteilen

Band 2:
1. Einleitung
2. Was Sie benötigen
3. Starten von APPLE WORKS
4. Der Schreibtischmanager
5. Textbearbeitung
6. Datenfernübertragung

A1...A6 wie Band 1, dazu:
A7 Modemkabel für APPLE IIe, IIc
A8 SuperSerialCard: Einstellung
A9 ASCII-Textdateien aus anderen Dateiformaten für ACCESS II
A10 DATEX-P20 F Verzeichnis
A11 Deutsche/Englische Menübilder von ACCESS II

APPLE WORKS auf APPLE II, IIe, IIc:
verwandelt APPLE-II-Computer in einen Elektronischen Schreibtischmanager mit:
Texterstellung ...Edition, Briefarchiv, Ausdruck etc. Datenarchivierung ...Kontoführung, Buchhaltung etc. Formblattkalkulation ...Bilanzen, VisiCalc-Dateien etc. Datenfernübertragung ...Mailbox, Rechnerkopplung etc.

- ist ein erfolgreicheres Integrationspaket als LOTUS auf IBM PC!
- ist auf 1 MByte Speichererweiterungen Ihres APPLE II vorbereitet!
- erschließt Ihnen die Zukunftstechnik MAILBOX!
- ist ebenso einfach zu bedienen wie APPLE WRITER:
  Kein Befehlsstudium ...Einfachste Menüführung ... Sofortige Anwendbarkeit

te-wi's APPLE WORKS SYSTEMBÜCHER 1+2 zeigen Ihnen:
- Sämtliche APPLE WORKS Funktionen an Beispielen aus der Wirtschaft
- Das Wechseln zwischen Text/Rechenblatt/Datenarchiv/Dfü
- Umfassende Systeminformationen zu Dateikonvertierung, Druckeranpassung etc.

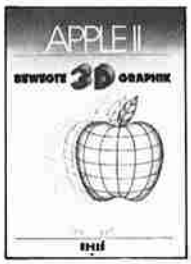
Von Botta/Lange/Zimmermann je 264 Seiten, Softcover, je DM 49,–

te-wi Verlag GmbH
Theo-Prosel-Weg 1
8000 München 40

Weitere te-wi-Bücher

Das APPLE II/II+/IIe/IIc-Handbuch NEU
(L. Poole)
Erst mit Hilfe dieses Leitfadens werden Sie Ihren Apple II erfolgreich einsetzen, denn Text und Bildmaterial gehen weit über das hinaus, was herstellerseitig an Literatur angeboten wird.
Neu überarbeitet und jetzt um die spezifischen Eigenheiten der Modelle **II e** und **II c** erweitert. 472 Seiten, Softcover, DM 66.–

LOGO – Jeder kann programmieren
(Daniel Watt)
Buch des Jahres in den USA. Für die Computer APPLE II, C-64, IBM PC, ATARI bis 520 ST., TI-99 und Schneider CPCs.
Hochwertiges Textbuch für Logo-Kurse zu Hause und im Lehrbereich.
384 Seiten, A4, DM 59.–

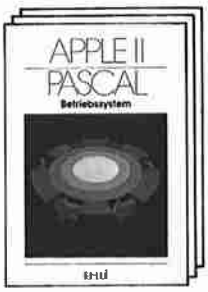


NEU

APPLE II – Bewegte 3D-Graphik
(Phil Cohen)
Selbstentworfene Graphiken und Diagramme – animiert oder als Standbilder – eben oder räumlich: alle erforderlichen BASIC-Programme mit Erklärung finden Sie in diesem Buch.
200 Seiten, Softcover, DM 49.–

Apple Maschinensprache
Für BASIC-Programmierer der einfachste Zugang zur Muttersprache des APPLE. Wesentlich schnellere Maschinenprogramme, direkte Manipulation des Mikroprozessors 6502 im APPLE – als Brücke dorthin benötigt dieses Buch nur die drei BASIC-Befehle POKE, CALL, PEEK. D. Inman/K. Inman. DM 49.–

Reparaturanleitung Computer: Apple II, IIplus NEU
Einzigartige Serviceunterlage für Reparaturen und Entwicklungsarbeiten am Apple II. Enthält Schaltpläne, Bauteile- und Vergleichstypenliste; Prüfpunkte mit Oszillogrammen der Signalformen, Logiktabellen, Spannungsangaben; schnelle Servicetests; Anleitung zur systematischen Fehlersuche.
In A4-Mappe, DM 29.80



Erstes deutsches Referenzwerk sämtlicher Befehle und Systemroutinen von Apple II, IIplus, IIe
APPLE II PASCAL NEU
Betriebssystem, 272 S., DM 49.–
Sprache, 216 S., DM 39.–
Pascal 1.2 Addendum, 112 S., DM 36.–

Grundlagenbuch, Bestseller APPLE II PASCAL.
Eine praktische Anleitung.
544 S., DM 59.–

tm 4422

Noch im Programm:
Computer für Kinder, APPLE II, DM 29,80
6502 Programmieren in Assembler, DM 59,–
Umweltdynamik (Prospekt anfordern), DM 66,– (NEU)
Macintosh Programmierhandbuch mit MSBASIC 2.0 (Ende '85), DM 59,–
Einführung in die Mikrocomputer-Technik, DM 66,–
M68000-Familie, 2 Bände, DM 79,– und DM 69,–

ProDOS-Uhrkartentreiber

Realtime-Clock „TIME II“

von Dipl.-Ing. Achim Sucker

Haben Sie sich auch schon geärgert, daß die neu erworbene Clockkarte nicht mit ProDOS zusammen arbeitet? Hier ist ein Programm, das in das STARTUP-Programm von ProDOS eingefügt werden kann. Getestet wurden die Versionen 1.0, 1.0.1 und 1.0.2. Das Programm ist auch allein lauffähig. In Zeile 1460 muß dann END stehen.
Im Original-STARTUP der ProDOS-Systemdiskette werden die Zeilen 170-1460 durch die neue Routine ersetzt. Auf diese Weise können Uhrzeit und Datum direkt nach dem Bootvorgang aus der Uhrenkarte eingelesen und in die ProDOS-Global-Page (Speicherstellen 49040-49043 = $BF90-$BF93) geschrieben werden. Bei jedem SAVE-Vorgang erscheint jetzt nicht mehr <NO DATE>, sondern das aktuelle Datum und die Uhrzeit im Directory.
Auf der Peeker-Sammeldisk #19 ist ein EXEC-File (CLOCK.EXEC), der in das STARTUP-Programm das Clockprogramm einfügt. Vorgehensweise zum Einfügen:

```
- LOAD STARTUP
- UNLOCK STARTUP
- SAVE STARTUPALT
- DEL 17Ø,146Ø
- EXEC CLOCK.EXEC
- SAVE STARTUP
```

Aufbau und Funktionweise

Aufruf über „T - Display / Set Time“ des STARTUP-Menüs. Das Programm ist für die Clockcard „Time II“ geschrieben, eine verbreitete Version auf Basis des Uhren-ICs MSM 5832. Die Anpassung an verschiedene Slots ist durch Änderung der Variablen S = Slot in Zeile 330 möglich.
Die Clockkarte hat 13 Register, die über die Adressen: A = $C082 + Slot * $10 bzw. B = A - 1 angesprochen werden. Mit „POKE B,16“ (Zeile 340) wird die Anzeige der Uhr „eingefroren“ (wichtig z.B. bei Stundenwechseln). Über „POKE A, Register-Nr. + 32“ werden die Register ausgelesen, die Ziffer steht im Feld „X(0...12) = Peek (A)“. Die Uhranzeige wird mit „POKE B,0“ wieder freigegeben.

```
Register  Ø - Sekunden Einer
Register  1 - Sekunden Zehner
Register  2 - Minuten Einer
Register  3 - Minuten Zehner
Register  4 - Stunden Einer
Register  5 - Stunden Zehner
Register  6 - Wochentag
              (Ø = Samstag,
               1 = Sonntag usw.)
Register  7 - Datum Einer
Register  8 - Datum Zehner
Register  9 - Monat Einer
Register 1Ø - Monat Zehner
Register 11 - Jahr Einer
Register 12 - Jahr Zehner
```

außerdem:

```
Bit 2 von Stunden Zehner:
      Ø = AM,
      1 = PM
Bit 3 von Stunden Zehner:
      Ø = 12-Std.-Format
      1 = 24 -Std.-Format
Bit 2 von Datum Zehner:
      Ø = kein Schaltjahr
      1 = Schaltjahr
```

Aus der Jahreszahl wird ein eventuell vorhandener 29. Februar berechnet und das entsprechende Bit auf der Uhrenkarte gesetzt.
Die Ausgabe auf dem Schirm:

```
  SAT--  2Ø-JUL-85
  TIME : 18:23:5Ø
UPDATE DATE AND TIME ?
```

Gegebenenfalls können Uhrzeit und Datum neu eingegeben werden, wobei der Wochentag in den Zeilen 1230-1270 neu berechnet und in der Uhr gespeichert wird. Die Formel gilt bis zum Jahr 2000. Da ProDOS 1984 geschrieben wurde, sind nur Jahreszahlen von 84...99 → 1984...1999 zulässig.
Das Beschreiben der Register (Zeile 1290 - 1320) erfolgt über:

```
POKE A, Register-Nr.
POKE B, Ziffer + 16
POKE A, Reg. + 16
POKE A, Reg.
```

Im Programm wird automatisch das 24-Stunden-Format gesetzt. Ist keine Uhrenkarte in den Slots vorhanden, wird

```
DATE : <NO DATE>
TIME : <NO TIME>
```

ausgegeben. Uhrzeit und Datum können wie herkömmlich eingegeben werden. In den Zeilen 1390-1450 werden in der Global-Page Datum und Uhrzeit aktualisiert.

Kurzhinweise

1. Zweck:
Programm zur Anpassung des ProDOS-STARTUP-Programms an die Uhrenkarte TIME II
2. Konfiguration:
II+ (mit LC) oder IIe;
Uhrenkarte TIME II;
ProDOS;
3. Test:
Diskette mit neuem STARTUP booten
4. Sammeldisk:
CLOCK
(Applesoft-Programm-Modul)
CLOCK.EXEC
(EXEC-Datei zum Patchen des Original-STARTUP)
Diese Dateien müssen zunächst mit CONVERT oder DOSTOPRO auf Ihre Kopie der ProDOS-Systemdiskette konvertiert werden.

CLOCK

```
101 DIM X(12): REM Für Datum/Uhrzeit-Utility
200 HOME
210 PRINT "****************************************"
220 PRINT "*                                      *"
230 PRINT "*        Datum / Uhrzeit-Utility       *"
240 PRINT "*     - für Uhrenkarte 'TIME II' -     *"
250 PRINT "*    von Dipl.-Ing. A. Sucker, 7.85    *"
260 PRINT "****************************************"
270 MO$ = "JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC"
280 DAY$ = "SATSUNMONTUEWEDTHUFRI"
290 REM
300 REM -Lies Uhrenkarte (Slot 5)
310 REM  Slot (S) = 5
320 REM
330 S = 5:A = 49282 + (S * 16):B = 49281 + (S * 16)
340 POKE B,16
350 FOR I = 44 TO 32 STEP - 1: POKE A,I:X(I - 32) =
    PEEK (A): NEXT
360 POKE B,0
370 IF X(8) > 3 THEN X(8) = X(8) - 4:
    REM BIT2 = 1 -- 29.FEB 'ON'
380 IF X(5) > 7 THEN X(5) = X(5) - 8:
    REM BIT3 = 1 -- 24 HR.FORMAT
390 IF X(5) > 3 THEN X(5) = X(5) - 4:
    REM BIT2 = 1 -- PM, 0 -- AM
400 YY = X(12) * 10 + X(11): IF YY < 84 OR YY > 99 GOTO 490
410 MM = X(10) * 10 + X(9):MM$ = MID$ (MO$,MM * 3 - 2,3)
420 DD = X(8) * 10 + X(7)
430 HH = X(5) * 10 + X(4)
440 MI = X(3) * 10 + X(2)
450 SS = X(1) * 10 + X(0)
460 DAY = X(6)
470 IF INT (YY / 4) = YY / 4 THEN X(8) = X(8) + 4:
    REM SET 29.FEB-BIT
480 POKE B,16: POKE A,8: POKE B,X(8) + 16: POKE A,24:
    POKE A,8: POKE B,0
490 REM
500 REM -Gib Datum/Uhrzeit aus
510 REM
520 VTAB 10: PRINT "        THE CURRENT DATE / TIME :"
530 VTAB 13: HTAB 14
540 IF YY < 84 OR YY > 99 THEN PRINT "DATE: <NO DATE>":
    PRINT :YY = 83: GOTO 580: REM keine Uhrenkarte in SLOT
550 PRINT MID$ (DAY$,DAY * 3 + 1,3);"-- ";
560 IF DD < 10 THEN PRINT "0";
570 PRINT DD;"-";MM$;"-";YY: PRINT
580 PRINT "                TIME: ";
590 IF YY < 84 THEN PRINT "<NO TIME>": GOTO 670
600 IF HH < 10 THEN PRINT "0";
610 PRINT HH;":";
620 IF MI < 10 THEN PRINT "0";
630 PRINT MI;":";
640 IF SS < 10 THEN PRINT "0";
650 PRINT SS
660 GOSUB 1390: REM Neues Datum/Uhrzeit
670 VTAB 19: PRINT "    UPDATE DATE AND TIME? Y/N ";: GET A$
680 HTAB 1: PRINT "                                  "
690 IF A$ = "N" OR A$ = "n" OR A$ = CHR$ (13) GOTO 1370
700 VTAB 13: PRINT "        ENTER DATE: DD-MMM-YY";
710 REM
720 REM -Tag nach DD
730 REM
740 HTAB 20: GET A$:DD = VAL (A$): PRINT A$;
750 GET A$:DD = DD * 10 + VAL (A$): PRINT A$;
760 IF DD < 1 OR DD > 31 THEN HTAB 20: PRINT "DD";: GOTO 740
770 REM
780 REM -Monat nach MM
790 REM
800 HTAB 23:MM$ = "": FOR I = 1 TO 3
810 GET A$: IF ASC (A$) > 96 THEN A$ = CHR$ ( ASC (A$) - 32)
820 PRINT A$;:MM$ = MM$ + A$: NEXT
830 FOR I = 1 TO 34 STEP 3
840 IF MM$ = MID$ (MO$,I,3) THEN MM = (I + 2) / 3:I = 38
850 NEXT
860 IF I < 38 THEN HTAB 23: PRINT "MMM";: GOTO 800
870 REM
880 REM -Jahr nach YY
890 REM
900 HTAB 27: GET A$:YY = VAL (A$): PRINT A$;
910 GET A$:YY = YY * 10 + VAL (A$): PRINT A$;
920 IF YY < 84 THEN HTAB 27: PRINT "YY";: GOTO 900
930 VTAB 19: HTAB 15: PRINT "CORRECT? Y/N ";: GET A$
940 HTAB 15: PRINT "              ": VTAB 13: HTAB 8:
    PRINT "      "
950 IF A$ = "N" OR A$ = "n" GOTO 700
960 VTAB 15: HTAB 8: PRINT "ENTER TIME: HH:MM    ";
970 REM
980 REM -Stunde nach HH
990 REM
1000 HTAB 20: GET A$:HH = VAL (A$): PRINT A$;
1010 GET A$:HH = 10 * HH + VAL (A$): PRINT A$;
1020 IF HH > 23 THEN HTAB 20: PRINT "HH";: GOTO 1000
1030 REM
1040 REM -Minute nach MI
1050 REM
1060 HTAB 23: GET A$:MI = VAL (A$): PRINT A$;
1070 GET A$:MI = 10 * MI + VAL (A$): PRINT A$;
1080 IF MI > 59 THEN HTAB 23: PRINT "MM";: GOTO 1060
1090 VTAB 19: HTAB 15: PRINT "CORRECT? Y/N ";: GET A$
1100 HTAB 15: PRINT "              "
1110 IF (A$ = "N") OR (A$ = "n") GOTO 960
1120 VTAB 15: HTAB 8: PRINT "      "
1130 REM
1140 REM -Uhrzeit und Datum der Uhrenkarte erneuern
1150 REM
1160 X(12) = INT (YY / 10):X(11) = YY - X(12) * 10
1170 X(10) = INT (MM / 10):X(9) = MM - X(10) * 10
1180 X(8) = INT (DD / 10):X(7) = DD - X(8) * 10
1190 X(5) = INT (HH / 10):X(4) = HH - X(5) * 10
1200 X(5) = X(5) + 8: REM BIT3 = 24-Stundenformat
1210 X(3) = INT (MI / 10):X(2) = MI - X(3) * 10
1220 X(1) = 0:X(0) = 0
1230 REM -Wochentag
1240 IF MM < 3 THEN F = INT ((1899 + YY) / 4)
1250 IF MM > 2 THEN F = - INT (.4 * MM + 2.3) +
     INT ((1900 + YY) / 4)
1260 F = F - 15 + 365 * (1900 + YY) + DD + 31 * (MM - 1)
1270 X(6) = F - INT (F / 7) * 7
1280 REM -Stelle Uhrenkarte
1290 POKE B,16
1300 FOR I = 12 TO 0 STEP - 1
1310 POKE A,I: POKE B,X(I) + 16: POKE A,I + 16: POKE A,I
1320 NEXT : POKE B,0
1330 REM -aktualisiere System-Datum und -Uhrzeit
1340 GOSUB 1390
1350 PRINT : VTAB 18: PRINT " * SYSTEM DATE AND TIME UPDATED *"
1360 PRINT : PRINT " *    CLOCKCARD TIME II SET     *"
1370 VTAB 23: PRINT " PRESS ANY KEY TO DISPLAY MAIN MENU ";:
     GET A$
1380 GOTO 1460
1390 REM
1400 REM -aktualisiere Datum und Uhrzeit
1410 REM
1420 POKE 49043,HH: POKE 49042,MI: POKE 49041,YY * 2 + (MM > 7)
1430 IF MM > 7 THEN POKE 49040,(MM - 8) * 32 + DD
1440 IF MM < 8 THEN POKE 49040,MM * 32 + DD
1450 RETURN
1460 GOTO 2540: REM Ende der Uhr-Utility
```

Datum und (!) Uhrzeit für „Appleworks"

Vorgehensweise:
ProDOS-Diskette mit Namen /APPLEWORKS formatieren

Folgende Files übertragen:
-- PRODOS und BASIC.SYSTEM von ProDOS-Systemdiskette
-- APLWORKS.SYSTEM und SEG.00 von Appleworks-Systemdiskette

CLOCK von Sammeldisk laden

Zeile 1370:
1370 PRINT CHR$(4) "-APLWORKS.SYSTEM"
ggf. können für einen automatischen Bootvorgang die Zeilen 670–1360 gelöscht werden.

Programm als STARTUP speichern

Nach jedem Appleworks-Bootvorgang auf die Abfrage des Datums mit <RETURN> antworten und fortan wird Datum **und** Uhrzeit im Directory vermerkt.

EDIT

Ein Luxus-Disk-Editor für ProDOS

von Arne Schäpers

Teil 2: Dumpen und Editieren

Der erste Teil von EDIT (siehe Peeker 5/1985, S. 8-25) mit der Beschreibung der Befehlsauswertung (ED.FRAME) und des PRODOS-Diskettentreibers (ED.DISKIO) ist dafür, was man mit ihm anfangen kann, reichlich lang geraten – über 1500 Zeilen Quelltext für ein Programm, bei dem man noch den Monitor benutzen muß, damit man überhaupt etwas von den gelesenen Daten zu sehen bekommt. Die hier vorgestellte Erweiterung von ED.FUNCS1 wird dem gründlich abhelfen, denn sie enthält die Funktionen „D"ump (Hex/ASCII), „A"SCII-Dump, „V"BM-Dump, „H"old, „C"ompare und „M"ove mit den entsprechenden Cursorsteuerungen und als wichtigste Funktion „S"et zum Verändern von Bytes im Arbeitspuffer namens BUFFER. (Die sich auf das Listing beziehenden Begriffe bzw. Labels sind in Großbuchstaben geschrieben.)

1. Beschreibung der Befehle

Der normale Prozeß zum Ändern von Bytes sieht so aus:
„D"ump des fraglichen Teils;
„S"et (= Ändern) der entsprechenden Bytes;
„D"ump zur Überprüfung der Änderung.
Für einen CP/M-Record mit 128 Bytes ist ein Dump nicht einmal mit einem 40-Z/Z-Bildschirm ein Problem – der gesamte Record läßt sich auf einmal darstellen. Bei einem DOS-3.3-Sektor mit 256 Bytes kommt man mit einem 40-Z/Z-Bildschirm bereits in Schwierigkeiten. Für einen ProDOS- oder Pascal-Block mit 512 Bytes ist selbst ein 80-Z/Z-Bildschirm unzureichend – ohne Scrollen oder Paging (seitenweise Unterteilung) geht nichts mehr.

Am einfachsten ist es natürlich, wenn man sich als Programmierer auf die Scroll-Funktion des jeweiligen Bildschirms verläßt – schließlich gibt es ja noch die Ctrl-S-Taste, mit der man den Ausdruck anhalten kann, wenn man schnell genug ist.

Der Programmieraufwand für ein starres Display ist ungleich höher, weil das Programm vor der Ausgabe feststellen muß, ob sich überhaupt ein geändertes Byte auf dem Schirm befindet. Wenn das Byte lokalisiert worden ist, wird es mit der neuen Information überschrieben. Wenn das Programm dann auch noch Hex und ASCII gleichzeitig ausgibt, muß zusätzlich die

Bildschirmstelle für den ASCII-Ausdruck lokalisiert und verändert werden. Für den Standard-Bildschirm des Apple (40 Z/Z) ist das noch möglich, doch an ein entsprechendes Programm für 80 Z/Z hat sich noch niemand herangetraut.
Nun habe ich ja in der Überschrift zu diesem Beitrag von „Luxus" gesprochen – deshalb beherrscht EDIT diese Darstellungsart für alle Konfigurationsmöglichkeiten gleichzeitig, für den IIc und IIe in 40 Z/Z, 80 Z/Z (Videx) und 80 Z/Z-Darstellung. Der eingangs erwähnte Prozeß des „D"ump/„S"et/„D"ump beschränkt sich also auf einen ersten „D"ump, gefolgt von „S"et und Ansehen.

1.1. Befehlsübersicht

Die neuen Funktionen haben fast alle denselben syntaktischen Aufbau. <Start,Ende> steht dabei für:

– keine Angabe: Start = 000, Ende = MAXBUF bzw. maximal mögliches Ende;
– „Start" angegeben, „Ende" nicht angegeben: „Ende" wird eingesetzt;
– „Start" und „Ende" angegeben – beide Werte werden berücksichtigt und überprüft, bei „Start" größer „Ende" erfolgt eine Fehlermeldung.

D<Start,Ende>: Hex- und ASCII-Dump (z.B. D000,100)
A<Start,Ende>: ASCII-Dump (z.B. A000,100)
V<Start,Ende>: VBM-Dump, d.h. in einzelnen Bits (z.B. V000,1FF)
H<Start,Ende>: kopiert Bytes aus dem Puffer in den Hold-Puffer (z.B. H000,0FF)
M<Start,Ende>: kopiert Bytes vom Hold-Puffer in den Puffer (z.B. M000,0FF)
C<Start,Ende>: vergleicht den Puffer mit dem Hold-Puffer (z.B. C000,0FF)

Die Cursor-Bewegungen innerhalb der Dump-Displays:
„<" und „,": halbe Seite abwärts bzw. Zeile abwärts
„>" und „.": halbe Seite aufwärts bzw. Zeile aufwärts

Die Funktion **Set**:
Sxxx:Hex Hex Hex ...: ändert Bytes ab xxx in Hex (z.B. S000:C1 C1 C2)
Sxxx'ASCII...: ändert Bytes ab xxx in ASCII, MSB (= höchstwertiges Bit = Bit 7) gelöscht (z.B. S000'ABC)
Sxxx" ASCII...: ändert Bytes ab xxx in ASCII, MSB gesetzt (z.B. S000"ABC)
Sxxx,yyy,zz: füllt von xxx bis yyy mit dem Byte zz (z.B. S000,1FF,AA)

1.2. Anmerkungen

Der durch „D", „A", „V" und „C" gesetzte Display-Modus bleibt bis zum nächsten Kommando dieser Art erhalten. Das bedeutet, daß jede Veränderung des Pufferinhaltes in dieser Art angezeigt wird, solange sich das entsprechende Byte momentan auf dem Bildschirm befindet.

Innerhalb dieser Modi wird durch die Cursor-Funktionen „umgeblättert". Die Cursorfunktionen sind „anreihbar", d.h. zwei „<" hintereinander ergeben eine Verschiebung um eine ganze Seite, drei „<" eine Verschiebung um 1 1/2 Seiten ohne einen erneuten Dump dazwischen. Analoges gilt für die drei anderen Cursor-Kommandos, nicht aber für eine Kombination daraus wie z.B. „<,". Hier wird erst eine halbe Seite umgeblättert und dann um eine weitere Zeile gescrollt.

Der Vergleich des Puffers mit einer vorher mit „H"old gespeicherten Anzahl von Bytes durch die Funktion „C"ompare setzt eine Ober- und Untergrenze für das Umblättern: Wenn zusammen mit „C"ompare eine Startadresse angegeben wird, wird das erste Byte im Puffer ab dieser Startadresse, ansonsten das Byte 00 in BUFFER mit dem ersten Byte des Hold-Puffers (HLBUF) verglichen. Ein Zurückblättern, bei dem diese Startadresse unterschritten wird, erzeugt natürlich sinnlose Ergebnisse und wird deshalb vom Programm nicht zugelassen; entsprechende Cursorbewegungen werden ignoriert.

Genauso verhält es sich, wenn das letzte Byte des Hold-Puffers erreicht ist. Ein Umblättern über diese Endadresse hinaus wird ebenfalls ignoriert („H" ohne Parameter hält den gesamten Puffer fest, „H" mit Start und Ende nur einen Teil davon; der Hold-Puffer ist dann kürzer als BUFFER).

Ein „M"ove ohne Parameter kopiert den gesamten Hold-Puffer, ein „M"ove mit angegebener Startadresse kopiert das erste Byte des Hold-Puffers auf diese Startadresse, das zweite auf die Startadresse+1 usw. Wird bei dieser Kopieraktion MAXBUF überschritten, bricht das Programm ohne Fehlermeldung ab.

Ein „M"ove mit angegebenem Start und Ende versucht, die entsprechende Anzahl von Bytes aus dem Hold-Puffer zu kopieren. Ist der Hold-Puffer vorher zu Ende, wird ebenfalls ohne Fehlermeldung abgebrochen. In beiden Fällen wäre ein Sprung zu CMDERR ohne weiteren Aufwand möglich gewesen, doch wollte ich dem Benutzer das manuelle Abzählen der HLBUF-Länge ersparen.

Kommen wir zur Funktion „S"et:
„S"et funktioniert noch auf eine weitere Art, nämlich ohne angegebene Startadresse vor dem Argument:
S: Hex Hex... bzw. *S'ASCII...* und *S"ASCI...* machen da weiter, wo das letzte Set aufgehört hat. Wenn Sie also fortlaufende Bereiche eines Blocks verändern wollen, entfällt das Abzählen der bisher veränderten Bytes innerhalb einer Zeile (was man normalerweise mit dem Finger auf dem Bildschirm macht und dabei murmelt: „8..9..A..B..C").

Im Gegensatz zu „M"ove bricht Set mit der Meldung „Out of Buffer Range" ab, wenn das Ende des Puffers erreicht wird. Selbstverständlich werden vor diesem Abbruch die noch veränderten Bytes auf dem Schirm angezeigt (es sei denn, Sie arbeiten im Blindflug: Display von $000..$0FF und „S"et der Bytes $1F0 und folgende – dann sehen Sie nur die Fehlermeldung, und der Cursor steht auf dem ersten Byte, das nicht mehr in den Puffer geschrieben werden konnte).

Die Funktion Set (ASCII) hat einen Haken: Bei einer Kommandozeile, in der mehrere Kommandos hintereinander stehen, müssen diese Kommandos in irgendeiner Form voneinander getrennt werden. Nach längerem Nachdenken habe ich mich dazu entschlossen, das Ausrufezeichen („!") dazu zu benutzen. Die Folge davon ist natürlich, daß dieses Zeichen selber innerhalb von Set nicht direkt benutzt werden kann und deshalb im Stil von Pascal als „!!" geschrieben werden muß.

2. ED.FUNCS1

2.1. Allgemeine Beschreibung

EDFUNCS1 umfaßt zusätzlich zu den oben besprochenen Funktionen noch die bereits im ersten Teil dieser Serie erläuterten Funktionen „X" (Monitor) und „Q" (Ende). Dieser Programmteil ist mit den neuen Erweiterungen beendet und wird nicht mehr verändert.

Für alle Dump-Funktionen existiert ein Satz zentraler Variablen, die in GLOBALS (in ED.FRAME) bereits definiert sind:

DSPBTM und **DSPTOP** – Absolute Unter- und Obergrenzen des im jeweiligen Modus darstellbaren Bereiches. Für alle Modi außer „C"ompare ist DSPBTM = 000 und DSPTOP = MAXBUF. Bei „C"ompare wird DSPBTM durch die Startadresse, DSPTOP durch Startadresse plus HLBUF-Länge (limitiert auf MAXBUF) gesetzt.

DSPSTART und **DSPEND** – Start- und Endadresse des Displays. DSPSTART wird durch die Cursor-Bewegungen im Bereich DSPBTM...DSPTOP verschoben.
SADDR und **EADDR** – Start- und Endadresse des jeweiligen Kommandos. Wenn das Kommando einen neuen Dump-Modus setzt, wird DSPSTART =

peeker Börse

Verkauf Hardware

SCHWEIZ: Apple II europlus orig. mit 2 orig. Apple-Disk. Epson MX 80 Typ III und orig. Monitor II. VHB Fr. 2500,–.
Apple II+ komp. mit 2 orig. Apple Disk-II und orig. Monitor sowie 16K-Karte, 256 RAM-Disk, 80 Zeichen-Karte (Videx), 2 parallele Interfaces, CP/M-3.0-Karte (ALS) mit Software. VHB Fr. 2200,–
Matthias Meyer, Haselstrasse 15. CH-9013 St. Gallen. Tel. tagsüber (071) 278511, abends 277737.

Apple IIe, 128KB, Z80, Philips-Monitor 22MHZ, 2 Laufwerke (Slimline), Orig. Turbo-Pascal 3.0 + Literatur *DM 2800,–*.
Tel. 06732/5463.

Apple IIe, 128K, 80Z + Monit. + 2 LW + Imagew. + SSC + Zubehör für DM 4400 abzugeb. teilw. auch einz. Tel. 06231/7945.

Apple IIe (orig.), 128KB, Mon. III, 2. LW, für DM 1700,–.
Tel. 07022/64465.

II + komp. 64K + 80Z/Z+Z80 + Sprachkarte + Monit. + Barcode-Leser + Joyst. + 2 LW + Drucker + Unmenge Literatur + Softw. Preis VHB.
Götz, Tel. 07422/53721.

Apple-II/voll kompatibel, 64KB, Z80-CPU (CP/M), 140/640KB-Drives, 80 Zeichen-Karte, 12"-Monitor, Matrix-Drucker, Programm. Tastatur (Operator), viel Software, VB 2000,–. Zelewski, Tel. 0221/4703739 (Köln).

Apple komp. 64K, 80Z, 2 Drives. DM 1000,–. T. 07071/45071 ab 18.00 Uhr.

Verk. Apple IIe komp., Monitor, Lit. NP 1650,– VB 1350,–. Tel. 07082/8876.

Macintosh 128 auf 512KB DM 299,–. Fa. Schlösser, Tel. ab 17.00 Uhr, 089-985889

Original Apple II Euro + mit Umfangreichem Zubehör, nähere Information. Tel. 06181/21364.

Gepard-Computer (Peeker 5/86) 512K, 10MHz 68000 + Software und Handbuch. T. 05407/6653.

AP-20 (IBS 68000, 128K) + Softw. DM 600,–. Bauer, Telefon 07121/40518.

Verkauf Software

IIe+c: CAD paint: exakte Zeichnungen m. Beschriftung (2 Stärken), 40 Funktionen, 80 fertige Symbole für Wohnungspläne + elektr. Schalt. Info 2 DM, Briefm., Disk: DM 99,– Scheck. Lohmann, M.-Müller-Ring 7, 6500 Mainz

*** **DOS 3.3 Disk Utility** ***
Kopierschutz, Hellofile (BIN, TEXT, ASOFT) Katalogschutz, CRTL c S. keine Eingaben ins DOS 3.3 bzw. MC o. ASOFT usw. das alles für DM 89,–! bei Frank Pickel, Hollersborn 27, 5410 Höhr-Grenzhausen. Tel. 02624/7602 **********

DISKETTEN **********
5 1/4" 48 tpi, DM 1,99
3 1/2" 135 tpi, DM 4,65
auch andere, 6 Monate Garant.
Allg. Austro-Ag., Ringstr. 10, D-8057 Eching, Tel. 08133/6116

Apple-II: Frei-Prg. Liste anf. b. Fa. Josef Gebhardt, PF 1174, 8458 Sulzbach-Rosenbg., T. 09661/52916.

Dazzle Draw DM 80,– + Intus-Priv.-Buchhaltg. DM 100,– für Apple IIe/c, Tel. 05209/3203 ab 18 Uhr.

Software Uhr für Apple II+, e, c, Zeitschaltmöglichkeit Diskette + Anleitung DM 25,– Oecking
Tel: Do. 0231/391920

Window-Technik für Applesoft nur DM 35,– (alle Apple II-Typen). E. Heinz, Püttkampsweg 13, 2000 Hamburg 52

Uni SOFT

SHOW JT! Business Graphics	89,–
Regressjon XVII	79,–
Cluster Analyse IX	79,–
Critical Path Method	79,–

Books & Software (09571) 3182

Fontrix Grafik-Textprogr. orig.!
ungebr. VHB Götz, Tel. 07422/53721.

Ankauf Software

Suche Software CP/M 80
Suche eine Kopie von der Original Disk. CP/M 80 56K. Vers2. 20B von Microsoft. – Gegen Bezahlung. CH-0041/41 5550, 77 Luzern.

Hausverwaltung-Programm für Apple IIe. Kallmeyer, Seegefelder 75, 1000 Berlin 20.

Suche Spiele für Apple IIe, Tel. 02204/68231.

Suche Init/Treiber usw. Für BASIS 256K Ramkarte/IIe/UCSD PASCAL 1.2 (evtl. auch DOS 3.3). Ingram, Lütticherstr. 30, Köln 1.

CP/M 2.2 Systemdiskette gesucht für IIe A. Grün Telefon 02151/302177

Handwerkerprogramm/Schreiner auf IIe. W. Müller, Sandhäuser Str. 1, 6900 Heidelberg, Tel. 06221/781132.

Ankauf Hardware

Abrechnungssoftware (Krankenkassen) **von der KBV zugelassen,** Lightpenbedienung gesucht. Zahle bis DM 4000,–. Tel. 02641/1441.

Suche Apple III-System.
Tel. 02641/1441.

Suche – defekte – oder – nicht mehr gebrauchte – Computer und Peripherie.
Chiffre Nr. P 1008

Apple II Graphics Tablet gesucht bis DM 1000,–.
Tel. 030/211 41 72

Verschiedenes

APPLE REPARATUREN
(auch compatible M-boards, z.B. **Atlas, Arca, CES, Datastar, Dipa, Lasar, Mewa, PC-48 + 64, Plato, Radix, o. ae.) sowie Zusatzkarten und Disk-Drives** führt unser Spezialistenteam mit mehr als 5-jähriger Kunden- und Reparatur-Dienst-Erfahrung, garantiert zuverlässig und besonders kostengünstig aus.
Bitte genaue Fehlerangabe sowie Tel. Nr. für evtl. Rückfragen nicht vergessen.
Auf Wunsch Kostenvoranschlag.
aaa-electronic gmbh
Habsburgerstr. 134, 7800 Freiburg, Tel. 0761/276864, Tx. 772642aaad

!!!! Da gibt's was umsonst !!!!
4 × im Jahr den neuen Katalog. Bühler Elektronik, Postfach 32, 7570 Baden-Baden

Apple IIe-64KB, Monitor II, DuoDisk-Laufwerk, Superserielle Karte, 80Z Karte. Incl. Software z.B. Apple Writer II, Quick File und vieles mehr sowie Handbücher.
Neupreis ohne Software DM 4300,–
Verkaufspreis komplett DM 2700,–
Tel. 08389/256

Systemwechsel Superpreise alles neuwertig! Accelerator IIe 450,–, Word-Juggler (Dt. Textv. incl. Wörterb.) 450,–, IIc 1700,–, Imagewriter 800,–, 10 MB-HD 2500,–; Gernot Eiler,
Tel. 0043-05513/67083

Ihre Erphi-Vertretung für die Schweiz: Beltronic, Im Chapf, 8455 Rüdlingen, Tel. 018673141, Telex 825981.

PASCMAK: Suche Vertriebspartner und Interessenten für mein neues Immobilienmakler-Programm. Heinz, Waldgürtel 7, 5060 Berg. Gld. 1

Für weitere Informationen zu einem der in dieser Ausgabe vorgestellten Produkte stehen Ihnen die Produktkarten zur Verfügung.

Bitte verwenden Sie für Kleinanzeigen die vorgedruckten Antwortkarten in diesem Heft.

Anzeigenschluß für Peeker 8/86 ist am 1.7.86

Für Ihre Unterlagen

Abonnement bestellt

am: _____

Vertrauensgarantie:
Ich habe davon Kenntnis genommen, daß ich die Bestellung schriftlich durch Mitteilung an den Dr. Alfred Hüthig Verlag GmbH, Postfach 10 28 69, 6900 Heidelberg innerhalb von 7 Tagen widerrufen kann. Zur Fristwahrung genügt die rechtzeitige Absendung des Widerrufs (Datum des Poststempels).

Peeker
Leserservice
Postfach 10 28 69
6900 Heidelberg

Für Ihre Unterlagen

Folgende Bücher bestellt:

_____

_____

_____

_____

_____

am: _____

bei:

Peeker
Versandbuchhandlung
Postfach 10 28 69
6900 Heidelberg 1

Für Ihre Unterlagen

Folgende Disketten
und Programme bestellt:

_____

_____

_____

_____

_____

am: _____

bei:

Peeker
Softwareabteilung
Postfach 10 28 69
6900 Heidelberg 1

peeker

Abo-Karte

Ja, ich möchte **Peeker** abonnieren.
Liefern Sie mir **Peeker** ab Ausgabe ..... zum Jahresbezugspreis von z. Zt. DM 72,– (Inland) inkl. MwSt. Die Lieferung erfolgt frei Haus. Porto, Verpackung und Zustellgebühren übernimmt der Verlag. Der Jahresbezugspreis für das Ausland beträgt z. Zt. DM 72,– plus DM 18,– Versandspesen.

X _____

Datum — 1. Unterschrift

Bitte lesen!

Vertrauensgarantie: Ich habe davon Kenntnis genommen, daß ich die Bestellung schriftlich durch Mitteilung an den Dr. Alfred Hüthig Verlag GmbH, Postfach 10 28 69, 6900 Heidelberg innerhalb von 7 Tagen widerrufen kann. Zur Fristwahrung genügt die rechtzeitige Absendung des Widerrufs (Datum des Poststempels).

X _____

Datum — 2. Unterschrift

Verlagshinweis: Das Abonnement verlängert sich zu den jeweils gültigen Bedingungen um ein Jahr, wenn es nicht 2 Monate vor Jahresende schriftlich gekündigt wird.

Wir können nur Bestellungen mit zwei Unterschriften bearbeiten.

peeker

Buch-Karte

Bitte senden Sie mir gegen Rechnung folgende Bücher:

- ☐ Bühler, Applesoft-BASIC, 3-7785-1094-0, DM 38,–
- ☐ Eggerich, dBase II, Bd. 1, 3-7785-1147-5, DM 39,80
- ☐ Eggerich, dBase II, Bd. 2, 3-7785-0987-X, DM 39,80
- ☐ Eggerich, dBase II, Bd. 3, 3-7785-0988-8, DM 39,80
- ☐ Gabriel, Applewriter, 3-7785-1234-X, DM 35,–
- ☐ Hagenmüller, Microsoft-BASIC, Bd. 1, 3-7785-1038-X, DM 38,–
- ☐ Juhnke/Redlin, Apple Pascal, Bd. 1, 3-7785-1246-3, ca. DM 40,–
- ☐ Kehrel, Apple Assembler lernen, Bd. 1, 3-7785-1151-3, DM 38,–
- ☐ Kehrel, Apple Assembler lernen, Bd. 2, 3-7785-1170-X, DM 38,–
- ☐ Schäpers, ProDOS Analyse, 3-7785-1134-3, DM 68,–
- ☐ Schäpers, Bewegte Apple-Graphik, 3-7785-1150-5, DM 58,–
- ☐ Stiehl, Apple DOS 3.3, 3-7785-1297-8, DM 28,–
- ☐ Stiehl, Apple ProDOS, Bd. 1, 3-7785-1098-3, DM 28,–
- ☐ Stiehl, Apple ProDOS, Bd. 2, 3-7785-1036-3, DM 30,–
- ☐ Stiehl, Apple Assembler, 3-7785-1047-9, DM 34,–
- ☐ Wassermann, Apple IIc Handbuch, 3-7785-1157-2, DM 35,–

_____

Datum — Unterschrift

peeker

Software-Karte

Bitte senden Sie mir gegen Rechnung folgende Disketten:

- ☐ Peeker-Sammeldiskette, einzeln
  Disk# _____, Disk# _____
  Disk# _____, Disk# _____
  Peeker je Disk DM 28,– (einzeln)
- ☐ Peeker-Sammeldiskette, im Fortsetzungsbezug ab Disk# _____ (Mindestbezug 6 Disketten) Preis je Disk DM 20,–
- ☐ Apple DOS 3.3, Begleitdisk., DM 28,–
- ☐ ProDOS, Band 1, Begleitdisk., DM 28,–
- ☐ ProDOS, Band 2, Begleitdisk., DM 28,–
- ☐ Apple Assembler, Begleitdisk., DM 28,–
- ☐ ProDOS-Editor 1.0, Programm, DM 98,–
- ☐ MMU 2.0, Programm, DM 98,–
- ☐ INPPUT 2.0, Programm, DM 98,–
- ☐ Softbreaker 1.0, Programm, DM 48,–
- ☐ DB-Meister, Programm, DM 290,–
- ☐ Superquick, Programm, DM 48,–
- ☐ Turtle Graphics, Programm, DM 98,–
- ☐ Disk 40, Programm, DM 48,–
- ☐ Kyan-Pascal 2.0, Programm, DM 170,–
- ☐ Kitchen Fast-Writer, DOS 3.3, DM 128,–
- ☐ Fast-Writer, ProDOS, DM 128,–
- ☐ Double-Hires-Tools für Applesoft, DM 28,–

_____

Datum — Unterschrift

Abo-Karte

Name

Firma

Straße

PLZ/Ort

Ich wünsche jährliche Berechnung durch:
☐ Verlagsrechnung ☐ Abbuchung von meinem Bank- bzw. Postscheckkonto

Bank/PschA

Bankleitzahl

Kto.-Nr.

Buch-Karte

Karte bitte vollständig ausfüllen

Vorname, Name

Firma

Straße

PLZ/Ort

Telefon mit Vorwahl

Software-Karte

Karte bitte vollständig ausfüllen

Vorname, Name

Firma

Straße

PLZ/Ort

Telefon mit Vorwahl

POSTKARTE

Peeker
Leserservice

Dr. Alfred Hüthig Verlag GmbH

Postfach 10 28 69

6900 Heidelberg

POSTKARTE

Peeker
Buchabteilung

Dr. Alfred Hüthig Verlag

Postfach 10 28 69

6900 Heidelberg 1

POSTKARTE

Peeker
Softwareabteilung

Dr. Alfred Hüthig Verlag

Postfach 10 28 69

6900 Heidelberg 1

INPUT 2.0

Ein Bildschirm-Maskengenerator für DOS 3.3 und ProDOS

von U. Stiehl

1984, Diskette und Manual, DM 98,–
ISBN 3-7785-1021-5

„Input 2.0" liegt wahlweise in der Bank 1 oder Bank 2 der Language Card und wird durch einen kurzen Driver in den unteren 48K aufgerufen.
Für jedes Feld der Bildschirmmaske lassen sich u. a. definieren: Feldlänge (bis zu 255 Zeichen) – Vtab – Htab – Datentyp (insgesamt 8 Typen) – Scrollflag (starre oder dynamische Maske) – Ctrlflag – Füllflag – Löschflag – Bildschirmflag (40- oder 80-Z-Darstellung). Innerhalb eines Eingabefeldes besteht jeder denkbare Redigierkomfort (Insert, Delete, Rubout, Restore usw.).
Gerätevoraussetzung: Apple IIe oder IIc; ferner Apple II+ im 40-Zeichenmodus

MMU 2.0

Memory Managements Utilities

für die Apple IIe 64K-Karte DOS 3.3 (und ProDOS)

von U. Stiehl

1984, Diskette und Manual, DM 98,–
ISBN 3-7787-1023-1

Insgesamt enthält die neue „MMU 2.0"-Diskette über 25 Programme, die neue Einsatzmöglichkeiten für die Extended 80 Column Card (erweiterte 80-Z-Karte = 64K-Karte für den Apple IIe) erschließen. Ein Teil der Programme laufen auch auf dem Apple II Plus, doch ist „MMU 2.0" primär für 64K-Karte-Besitzer gedacht.

Gerätevoraussetzung: Apple IIe mit 64K-Karte oder IIc

DISK 40

Disketten-Organisationsprogramm für DOS-3.3 - 35-40 Spuren

von Hermann Seibold und Dipl.-Ing. Udo Marin, 1986, Programmdiskette mit Anleitung, DM 48,–

Durch eine einfach zu bedienende Menüführung können DOS-3.3-Disketten umfangreich bearbeitet oder kopiert werden.
– Tabellarische Ausgabe der Diskettenbelegung
– Ordnen des Catalogs
– „Undelete"n von versehentlich gelöschten Dateien
– Vergleichen von Disketten, Dateien oder DOS-Spuren
– Kopieren von Disketten, Dateien oder DOS-Spuren
– Formatieren von Daten-Disketten
– Erweitern auf 40 Spuren bei bestehenden 35-Spur-Disketten
– Ändern des Boot-Programms
– File-Editor zum Editieren von Disketten-Dateien
– Komfortabler Sektor-Editor für Hex- und ASCII-Darstellung
–VTOC-Editor, z. B. zur Freigabe der DOS-Spuren

Hüthig Software Service, Postfach 10 28 69, D-6900 Heidelberg

Verlangen Sie mehr

Computerbücher aus dem Fachbuchverlag Hüthig:
Präzise Informationen, aktuelle Themen,
gut lesbarer gesetzter Text.

Arne Schäpers
ProDOS-Analyse
Versionen 1.0.1, 1.0.2, 1.1.1
1985, 470 S., kart., DM 68,—
ISBN 3-7785-1134-3
Dies ist die umfangreichste und detaillierteste Darstellung, die jemals ein Apple-Betriebssystem erfahren hat. So etwas gibt es nicht einmal in Amerika!

Arne Schäpers
Bewegte Apple-Grafik
1985, 305 S., 6 Abb., kart., DM 58,—
ISBN 3-7785-1150-5
Ein Kursus für alle, die auf dem Apple hochaufgelöste und bewegte Grafiken in Maschinensprache programmieren wollen. Schrittweise wird ein Arcade-Spiel entworfen, das käuflichen Action-Spielen in der meisterhaften Grafik als Vorbild dienen kann.

Frank Bühler
Applesoft BASIC
Tips und Tricks
1985, 241 S., 40 Abb., kart., DM 38,—
ISBN 3-7785-1094-0
Das Buch enthält eine komplette Beschreibung aller möglichen Applesoft-Befehle und zeigt an einem Beispiel die erforderliche Syntax auf. Ausgearbeitete Unterroutinen können leicht in eigene Programme übernommen werden.

Jürgen Kehrel
Apple-Assembler lernen
Band 1: Einführung in die Assembler-Programmierung des 6502
1985, 235 S., kart., DM 38,—
Noch nie war es so einfach, die Assembler-Programmierung auf dem Apple zu erlernen, wobei auch der neue 65C02-Prozessor behandelt wird.

Ulrich Stiehl
Apple Assembler
1984, 200 S., 3 Abb., kart., DM 34,—
ISBN 3-7785-1047-9
Alle wichtigen ROM-Routinen sowie eine Vielzahl von Hilfsprogrammen werden in diesem Buch für den Programmierer mit Anfängerkenntnissen vorgestellt. Insgesamt über 40 Utilities mit mehreren völlig neuartigen Programmen sind gelistet.

Ulrich Stiehl
Apple DOS 3.3
Tips und Tricks
2., durchges. Auflage 1984, 216 S., kart., DM 28,—
ISBN 3-7785-1049-5
Das Standardwerk zum DOS 3.3 sowohl für BASIC- als auch für Assemblerprogrammierer. Enthüllt einige bislang noch niemals publizierte Techniken und viele Tricks aus der langjährigen Praxis des Autors. Dieses Buch ist der unentbehrliche Begleiter für jeden Apple-Programmierer.

Ulrich Stiehl
ProDOS für Aufsteiger
Band 1
2., geänderte Auflage 1985, 208 S., kart., DM 28,—
ISBN 3-7785-1098-3
Das Schwergewicht des ersten Bandes liegt auf der Darstellung der Pro-DOS-internen Systemadressen und der Assemblerprogrammierung unter diesem neuen Betriebssystem. Das Werk enthält auf über 70 Seiten eigens hierfür entwickelte Programme.

Carl-Ulrich Wassermann
Apple IIc
Handbuch für Anwender und Programmierer
1985, 324 S., kart., DM 35,—
ISBN 3-7785-1157-2
Ein Handbuch für alle, die die besonderen Eigenschaften der Apple IIc voll ausschöpfen wollen.

Weitere Titel und Informationen finden Sie in unserem Computerbuch-Katalog:
Dr. Alfred Hüthig Verlag, Postfach 10 28 69, 6900 Heidelberg 1

Hüthig

SADDR und DSPEND = EADDR. Für Set und REDUMP nach einer Veränderung des Puffer-Inhalts werden DSPSTART und DSPEND für die Darstellung der veränderten Bytes nur temporär gesetzt und erhalten nachher ihren alten Wert zurück.

2.2. Problemstellungen

Das größte Problem ist der falsche Prozessor. Bedingt durch die Tatsache, daß der 6502 leider überhaupt keine 16-Bit-Operationen zuläßt (wie etwa der Z80) ergibt sich durch die Blocklänge von $200 Bytes eine schier endlose Folge von Überträgen, 16-Bit-Subtraktionen mit bedingten Sprüngen für einen einfachen Vergleich zweier Adressen usw.

Alle Dump-Funktionen sollten nicht einfach von einer gegebenen Startadresse zeilenweise „herunterradeln", sondern einen formatierten Ausdruck liefern, d.h. wenn die Startadresse nicht mit z.B. $x00, $x10, $x20... beginnt, sondern mit einem „krummen" Wert, etwa $23, dann sollte die erste Zeile von $23 bis $2F gehen, die folgenden Zeilen gehen dann „ordentlich" mit $30, $40 usw. weiter. Daraus ergibt sich natürlich dann hinterher ein zusätzliches Problem bei der Bestimmung der Bildschirmposition eines Bytes.

Die Routine **REDUMP**: Nach Veränderung von einem oder mehreren Bytes wird REDUMP aufgerufen. Hier muß zum einen bestimmt werden, ob sich zumindest eines der veränderten Bytes momentan auf dem Bildschirm befindet. Wenn nein, ist REDUMP damit auch schon beendet.

Am einfachsten wäre ansonsten ein REDUMP des kompletten Schirminhaltes – das dauert aber mit den 80-Zeichenkarten eine halbe Ewigkeit. Folglich wird die erste Bildschirmzeile berechnet, in der eine Veränderung stattgefunden hat. Eine Bestimmung des Horizontal-Zeigers zu diesem ersten veränderten Byte ist aufgrund der verschiedenen Dump-Modi extrem aufwendig; deshalb wird dann über DUMP die gesamte Zeile neu gedruckt. Sind noch weitere Bytes in folgenden Zeilen verändert worden, werden diese Zeilen ebenfalls komplett neu gedruckt.

Die Funktionen „M"ove und „C"ompare benötigen eine doppelte Limitierung: die Endadresse wird zum einen durch die Startadresse plus der Länge des Hold-Puffers bestimmt, zum anderen darf sie MAXBUF nicht überschreiten.

2.3. Ausgewählte Routinen

XMON und **QUIT** (Z. 1-60) sind bereits im ersten Teil besprochen worden.

HALFUP/LINEUP (Z. 62-91) werden durch „<" bzw. „." aufgerufen. Da ein Display-Feld insgesamt 16 Zeilen enthält, unterscheidet sich HALFUP von LINEUP nur durch eine zusätzliche Multiplikation mit dem Faktor 8.
Beide Funktionen prüfen zuerst, ob noch weitere gleiche Kommandos folgen, d.h. „<<<" wird in einem einzigen Schritt als „3 mal <" berechnet. Die Routine XN BYTES multipliziert danach das durch „<" oder „." bzw. deren Vielfache erzeugte Argument mit der Anzahl der Bytes pro Zeile im momentanen Dump-Modus (NBYTES). Das Ergebnis wird vom alten Display-Start (DSPSTART) abgezogen. Wird dabei ein Wert < 0 erreicht, wird auf 0000 korrigiert. TSTBTM (Z. 83) vergleicht danach, ob die absolute Untergrenze (nur bei „C"ompare kann sie ungleich 0 sein) unterschritten wurde und korrigiert gegebenenfalls. Weiter geht es mit DOSCRL (Scroll-Ausführung).

HALFDN/LINEDN (Z. 96-121) führen denselben Prozeß durch wie HALFUP/LINEUP, nur mit umgekehrtem Vorzeichen und einem Test gegen DSPTOP, die absolute Obergrenze. Hier liegt ein Schönheitsfehler des Programms versteckt: Das Limitieren auf die Obergrenze allein ist nicht ausreichend, denn es soll schließlich noch etwas dargestellt werden. Folglich wird nach einem Limit einmal die „Anzahl Bytes/Zeile" (NBYTES) abgezogen. Danach muß geprüft werden, ob eventuell die Untergrenze (DSPBTM) unterschritten wurde. Das ist z.B. bei „Bytes/Zeile = 8, Länge Hold-Puffer: kleiner 8", „C"ompare und einem anschließenden Scroll-Versuch der Fall. Folge: Bei einem weiteren Scroll-Kommando in Richtung Puffer-Ende bewegt sich der Cursor eventuell eine halbe Zeile rückwärts, wenn das Puffer-Ende bereits erreicht ist.

DOSCRL (Scroll-Ausführung, Z. 124-132) setzt den neuen DSPSTART und gleichzeitig DSPEND auf die maximal mögliche Adresse, damit (soweit möglich) eine ganze Bildschirmseite gefüllt wird. Falls der Puffer zu Ende ist, bevor das Display-Feld vollständig beschrieben werden konnte, müssen danach die restlichen Zeilen durch den Einsprung in CLRDSP gelöscht werden.

MORE (Z. 135-144) prüft CMDSTR auf „mehrere gleichartige Zeichen" und liefert die Anzahl im Y-Register. Diese Routine wird nur für die Cursorsteuerungen benutzt.

MULT8 und **XNBYTES** (Z. 146-171) ist eine recht primitive 16-Bit-Multiplikationsroutine, die ebenfalls nur für die Cursorsteuerungen benutzt wird. Sie liefert das Ergebnis

SCLARG = X-Reg * A-Y, wobei Y der höherwertige Teil des Multiplikanden ist.

TSTMODE (Z. 175-181) prüft vor einem (RE)DUMP, ob irgendein Dump-Modus gesetzt ist und kehrt in diesem Fall mit gesetztem N-Flag („MI") zurück.

NOMODE (Z. 183-189) setzt alle Dump-Modi zurück und wird jeweils vor dem Setzen eines neuen Modus aufgerufen.

SAVEND und **RESEND** (Z. 192-206) speichern bzw. laden DSPEND vor bzw. nach einem REDUMP.

SADSTART (Z. 208-212) setzt DSPSTART auf die Startadresse SADDR und wird momentan nur beim Setzen eines neuen Dump-Modus benutzt.

REDUMP (Z. 215-289) ist reichlich kompliziert: Die Routine wird immer mit gesetzter SADDR und EADDR aufgerufen. REDUMP kehrt ohne Ausführung zurück, wenn eine der folgenden Bedingungen zutrifft:
– kein Dump-Modus gesetzt (TSTMODE);
– Bildschirm abgeschaltet (NODSP-Flag, ein Vorgriff auf Teil 3);
– EADDR (Adresse des letzten modifizierten Bytes) kleiner DSPSTART, d.h. keines der veränderten Bytes befindet sich auf dem Bildschirm;
– SADDR größer DSPEND, z.B. Modifikation eines Bytes im „C"ompare-Modus, das außerhalb des vergleichbaren Bereichs liegt, oder vorheriger Dump mit angegebener Endadresse: Die veränderten Bytes befinden sich nicht auf dem Bildschirm.
Ansonsten wird SADDR „normalisiert", d.h. mit einer trickreichen Aktion auf einen „Zeilenanfang" gesetzt, und mit DSPSTART verglichen. Wenn SADDR kleiner oder gleich DSPSTART ist, dann wird ab der ersten Display-Zeile bis EADDR ein neuer Dump ausgeführt.
Ist SADDR dagegen größer als DSPSTART, muß als nächstes die Zeile bestimmt werden, in der das erste modifizierte Byte steht. Der entsprechende Loop (NOTLIN0, Z. 255) beginnt mit der Bildschirm-Zeilennummer 6 und zieht solange „Bytes per Zeile" von der Differenz SADDR-DSPSTART ab, bis DSPSTART erreicht ist, während die Zeilennummer (Y-Register) laufend erhöht und die verbleibenden Zeilen im Display (DLINES) erniedrigt werden. Wenn DLINES gleich 0 wird, ist das Schirmende erreicht und ein Dump findet nicht statt. Ansonsten wird SADDR als nächste zu dumpende Zeile gesetzt, und EADDR wird auf den Wert „Ende der Zeile, in der das letzte modifizierte Byte steht + 1" erhöht, damit die ganze Zeile ausgedruckt wird. DSPEND wird zwischengespeichert und mit EADDR verglichen. Falls EADDR größer ist, wird auf DSPEND begrenzt. Die Routine SCPIDX, die in Z. 287 aufgerufen wird, setzt den entsprechenden Index zu

SADDR innerhalb des Hold-Puffers für den Fall, daß als Dump-Modus „C"ompare vorliegt. Und (endlich) wird in den Dump-Loop hineingesprungen, wo von NXTADDR...EADDR neu gedruckt wird. Nach einem Wiederherstellen des alten Wertes von DSPEND endet REDUMP mit einem RTS zur aufrufenden Routine.

ASCII/HEX/VBM-DUMP (Z. 296-314) setzen erst einmal einen „vorläufigen" NXTMODE und eine ebenfalls „vorläufige" Anzahl von Bytes pro Zeile (LBYTES), die auf der 80-Zeichendarstellung basieren. „Vorläufig" deshalb, weil im nächsten gemeinsamen Schritt GETPRM aufgerufen wird, denn hier kann ein Abbruch durch einen unzulässigen Parameter erfolgen. Der vorherige Dump-Modus bleibt dann unverändert. Wenn in GETPRM alles gut gegangen ist, werden DSPBTM auf 000 und DSPTOP auf MAXBUF gesetzt.

SETMOD2 (Z. 316-325) ist der Einsprung für das Setzen des vierten Dump-Modus, nämlich COMPARE. Hier sind DSPBTM und DSTOP je nach Startadresse und Länge von HLBUF innerhalb des COMPARE-Kommandos gesetzt worden. Die Endadresse (EADDR) wird auf DSPEND kopiert, alle Dump-Modi werden zurückgesetzt (NOMODE), und der neue Modus wird gesetzt.

SETDUMP (Z. 327-342) teilt die Anzahl der Bytes pro Zeile durch zwei, falls das Display auf einen 40-Z/Z-Bildschirm geht und nicht ASCII-Dump spezifiziert ist. In diesem Fall passen auch bei 40 Z/Z sämtliche Zeichen auf eine Zeile. Aus der Anzahl der Bytes pro Zeile wird für die formatierte Ausgabe die CRMASK gewonnen. Da alle Byte-Anzahlen reine Zweierpotenzen sind, ergibt sich bei einer Subtraktion von 1 ein Ergebnis, in dem alle Bits, die niederwertiger als die Byte-Anzahl sind, auf 1 gesetzt sind.

DUMP (Z. 345-359) wird auch noch von einigen (noch nicht implementierten) Routinen direkt angesprungen. Deshalb steht hier noch einmal ein Test mit dem Flag NODSP (Bildschirm aus). Ansonsten werden 16 Zeilen als Maximum gesetzt. Wenn der Dump zum Drucker geht (s. Teil 3), wird mit einem einfachen ORA mit dem PRINTER-Flag, das dann auf $FF gesetzt ist, diese Zahl auf $FF erhöht, d.h. es findet keine Begrenzung von DUMP durch den Zeilenzähler DLINES statt. Die nächste Adresse, ab der gedumpt werden soll, ist DSPSTART. Der Aufruf von SCPIDX setzt den entsprechenden Index innerhalb des Hold-Puffers für den Fall, daß es sich um einen COMPARE-Dump handelt. Sollte überhaupt kein Modus aktiv sein, so wird dies durch einen Aufruf von TST MODE festgestellt. DUMP ist ggf. damit beendet.

DLOOP (Z. 361-459) ist das Kernstück sämtlicher Modi: Nach dem Ausdrucken der Zeilen-Startadresse durch PRADDR wird via GNLINE eine Zeile aus dem Puffer „eingesammelt". GNLINE bricht dabei ab, wenn entweder DSPEND oder das Zeilenende, gesetzt durch CRMASK, erreicht ist. Die „eingesammelten" Bytes befinden sich danach im Speicherbereich LBUF, und der Index zum letzten Byte innerhalb dieses Puffers ist die Speicherstelle LTOP. Je nach Modus wird jetzt eine Zeile ausgedruckt, und zwar in Hex mit folgenden ASCII, nur ASCII, bitweise oder zuerst Hex. Alle diese Schleifen enden mit LINDON (Zeile fertig, Z.452). Eine Ausnahme stellt der VBM-Dump dar, denn hier muß erst geprüft werden, ob auf den 40-Z/Z-Bildschirm gedruckt wurde. Wenn dem so ist, befindet sich der Cursor nach dem Ausdruck von 4 Bytes bereits in der nächsten Zeile.
In LINDON folgt dann ein Clear End_of_Line, falls der Ausdruck an einer „krummen" Adresse endet. Danach folgt ein <CR>, das wegen der Unterschiede zwischen Videx- und IIc/e-80-Zeichenkarte in ED.FRAME definiert ist, wo die EDIT-interne Cursorposition auf den neuen Stand gebracht wird.
Der Zeilenzähler DLINES wird erniedrigt (Z. 454), wenn Null erreicht ist. Dann ist das Display-Feld voll und der Dump beendet.
Wenn GNLINE beim „Einsammeln" den Wert von DSPEND erreicht hat, ist das Flag DDONE gesetzt und der Dump zu Ende. Ansonsten steht NXTADDR (innerhalb von GNLINE kontinuierlich erhöht) auf dem Anfang zur nächsten Zeile, und es erfolgt so lange ein erneuter Sprung zu DLOOP, bis entweder der Zeilenzähler DLINES ausgelaufen oder DSPEND erreicht worden ist.

GNLINE (Z. 461-470) benutzt GNBYTE und sammelt maximal $20 Bytes (ASCII-Dump, sonst sind es weniger) in den Zeilenbuffer LBUF.

GNBYTE (Z. 473-493) zeigt, welche Verrenkungen mit einem reinen 8-Bit-Prozessor für das schlichte Holen eines Bytes mit einem Test auf Obergrenze notwendig sind, wenn der Quellbereich größer als $100 Bytes ist.
Hier wird zuerst die Quelladresse entsprechend justiert, danach 1 Byte aus dem Puffer geholt und zwischengespeichert. Danach werden „DSPEND erreicht" und „Zeilenende" geprüft und die Flags CLRF und DDONE entsprechend gesetzt, bevor die Routine mit dem Byte aus dem Puffer im Akku zurückkehrt.

GCBYTE (Z. 504-513) ist das Gegenstück zu GNBYTE. Hier werden die Bytes aus dem Hold-Puffer (HLBUF) geholt. Da die Begrenzung von DSPEND auf die HLBUF-Länge in den Routinen stattfindet, die DUMP aufrufen, muß hier kein Test auf HLBUF-Ende stattfinden.

SET (Z. 520-592) besteht eigentlich nur aus den bereits besprochenen Funktionen ohne irgendwelche Besonderheiten.
Wenn bei SET keine Startadresse angegeben ist, wird die interne Startadresse SETAD eingesetzt, die während des weiteren Verlaufs von SET kontinuierlich erhöht wird. SETAD dient auch beim abschließenden REDUMP als Endadresse (EADDR).

GETHEX (Z. 595-608) holt die nächste Hexzahl über EVALARG bzw. EVALHEX aus CMDSTR. Vorher werden eventuelle Leerzeichen übersprungen, die natürlich zum Trennen der Hexzahlen benötigt werden.

SBYTE (Z. 610-621) prüft vor dem Setzen eines Bytes, ob MAXBUF überschritten wurde und kehrt dann mit gesetztem Carry zurück, ohne das Byte zu setzen. Das führt dann innerhalb von SET zu BUFEXX, wo zuerst via SETQT die geänderten Bytes neu gedruckt werden. Danach wird dann die Fehlermeldung „Out of Buffer Range" ausgegeben. Der Cursor steht dabei auf dem ersten Zeichen, das nicht mehr in den Puffer gepaßt hat.

COMPARE (Z. 629-678) besteht von der Funktion her aus zwei Teilen, wobei der zweite Teil erst aus dem Listing in Teil 3 dieses Beitrages ersichtlich wird.

HLDERR (Z. 680-683) ist der Error-Exit für COMPARE und MOVE, falls HLBUF nicht gesetzt worden ist, und endet mit einem Sprung zu CMDERR nach Setzen des entsprechenden Zeigers innerhalb der ERRMSGS.

SCPIDX setzt den Index zum nächsten Vergleichsbyte in HLBUF durch Subtraktion der durch die COMPARE-Startadresse festgelegten CPBASE von der Adresse des nächsten Puffer-Bytes (NXTADDR).

HOLD (Z. 700-726) kopiert die gewünschte Anzahl von Bytes aus BUFFER nach HLBUF. Die Start- und Endadresse ist entweder angegeben oder von GETPRM mit 000 und MAXBUF gesetzt. HOLD verwendet dazu die Routine GNBYTE aus DUMP und setzt deshalb vorher die entsprechende NXTADDR. Da GNBYTE mit DSPEND endet, wird DPSEND zwischengespeichert und durch EADDR bis zum Ende der Kopieraktion ersetzt. Am Ende der Kopierschleife wird HLNUM (Anzahl der Bytes im

Festplattenkomplettlösung für jedermann

Mit der Firma Frank & Britting GmbH, die auf Festplatten spezialisiert ist, konnten wir ein extrem günstiges Sonderangebot aushandeln, das eine Festplattenkomplettlösung selbst für Apple-Besitzer mit kleinerem Geldbeutel erschwinglich macht. Sie können unter zwei Varianten wählen:

Luxus-Lösung: 20-Megabyte-Festplatte MDB20 (MDB = Mobile Datenbox) + Megaboard-Controller + Handbuch + 3 Konfigurationsdisketten + Anschlußkabel + DB-Meister-Dateiverwaltungsprogramm + Handbuch + 2 Programmdisketten zum Gesamtpreis von nur DM 3199,- inkl. MwSt.
Standard-Lösung: 10-Megabyte-Festplatte MDB10 + Megaboard-Controller + Handbuch + 3 Konfigurationsdisketten + Anschlußkabel + DB-Meister-Dateiverwaltungsprogramm + Handbuch + 2 Programmdisketten zum Gesamtpreis von nur DM 2799,- inkl. MwSt.
(jeweils 6 Monate Garantie.)

Wozu eine Festplatte?

Für eine Festplatte sprechen zwei Argumente, nämlich erstens der bedeutend größere Datenspeicher und zweitens die sehr hohe Zugriffszeit.
● 80- und 160-Spur-Laufwerke haben zwar einen größeren Datenspeicher als die normalen 35-Spur-Laufwerke, doch ist die Zugriffszeit hier wie dort bescheiden.
● RAM-Karten haben zwar eine hohe Zugriffszeit, doch ist der Datenspeicher flüchtig: Strom aus – Daten weg!
Für unsere Festplatten-Standard-Lösung (MDB10) gilt demgegenüber:
– Die Nettospeicherkapazität beträgt hier ca. 9.650K = ca. 9.880.000 Zeichen = ca. 15 Disketten mit 160 Spuren = ca. 69 Disketten mit 35 Spuren.
– Die Datenübertragungsrate beträgt auf unterster Ebene ca. 44K/s = ca. 45.000 Zeichen/s. Zum Vergleich beläuft sich die Übertragungsrate bei normalen Diskettenlaufwerken auf ca. 7,5K/s und bei RAM-Karten auf ca. 44K/s, d.h. RAM-Karten sind also keineswegs schneller, sondern meist sogar langsamer als die MDB-Festplatte.
Sie erhalten deshalb mit unserer Festplattenlösung einen großen externen Massenspeicher bei sehr hoher Datenübertragungsrate zu einem äußerst niedrigen Preis.

Unkomplizierte Installation

Die eigentliche Festplatte, also die „Mobile Datenbox" MDB, ist ein kleines rechteckiges Kästchen mit der Länge einer Handfläche. Das eine Ende des Flachbandkabels wird in die MDB, das andere Ende in den Megaboard-Controller gesteckt, der wie eine normale Erweiterungskarte aussieht und in jeden Slot von 1-7 paßt. Die MDB hat ein eigenes Netzteil. Deshalb brauchen Sie jetzt nur noch die MDB und danach den Apple einzuschalten, und fertig ist die Installation. Im Gegensatz zur Megacore-Festplatte, die wegen des Apple-Netzteil-Wechsels von Apple-Händlern eingebaut werden sollte, ist die Installation der MDB von jedermann problemlos zu bewerkstelligen.

Problemlose Konfigurierung

Danach booten Sie die mitgelieferte Systemdiskette DOS-CONFIG von einem 35-Spur-Laufwerk, tippen „I" für Initialisieren, geben die Festplattenparameter ein (bei MDB10: 4 Köpfe, 306 Zylinder, Präkompensation ab dem 256. Zylinder und Schneller Zugriff Ja), und nach zwei weiteren Kontrollbestätigungen beginnt die Initialisierung. Dies dauert ca. 14 Minuten für die MDB10, und nunmehr kann die Festplatte mit den Betriebssystemen beschrieben werden.

Als Betriebssysteme können Sie im einzelnen verwenden:
– DOS 3.3,
– Diversi-DOS 2C und 4C,
– Apple-Pascal 1.1 und 1.2,
– ProDOS 1.0.1, 1.0.2, 1.1.1,
– CP/M 2.2 56K (z.Zt nur diese CP/M-Version).

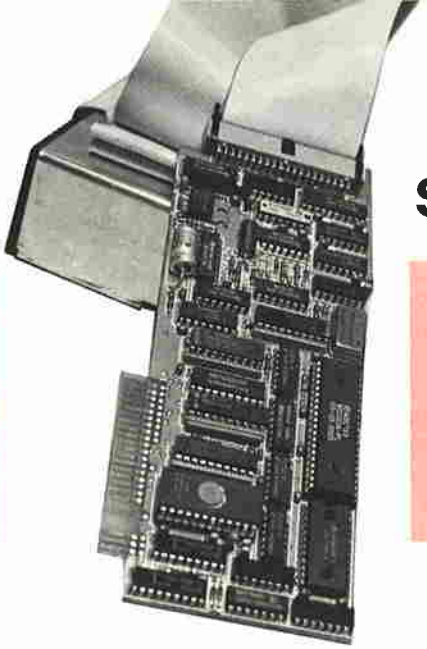
Die Konfigurierung, d.h. die Speicherverteilung auf die verschiedenen Betriebssysteme, sollte mit Bedacht überlegt werden. Nicht-benötigte Betriebssysteme können Sie natürlich weglassen. Nehmen wir an, Sie brauchen zunächst nur DOS 3.3. Wie wird nun konfiguriert?

Sie booten nach der obigen Initialisierung die Festplatte mit PR#7 und wählen aus dem automatisch erscheinenden Bildschirm-Menü „C" für „Configuration". Dann Tippen Sie „D" für DOS 3.3, danach „V" für Volumes, danach z.B. „10" für 10 Volumes = 2 x 10 x 140 = 2800K (1 Volume entspricht einem 35-Spur-Doppellaufwerk). Nun können Sie nach zwei weiteren Kontrollbestätigungen die neue Konfiguration auf die Diskette schreiben, was nur wenige Sekunden dauert. So einfach ist das! Später können Sie dann, sobald Sie mit der Festplatte besser vertraut sind, bei Bedarf die Konfigurationen für ProDOS, Apple-Pascal und CP/M nachholen.

Wie spricht man die MDB an?

Sie werden sich nun vielleicht fragen, wie Sie die Festplatte ansprechen müssen und ob Sie neue DOS-Befehle lernen müssen. Mitnichten! Booten Sie einfach die Festplatte mit PR#7 – übrigens heißt es immer PR#7, auch wenn Sie die Megaboard-Karte z.B. in Slot 4 gesteckt haben – und danach z.B.
INIT HELLO, S7, V3, D1
Damit wird das Volume in „Drive 1, Slot 7" formatiert. Nun können Sie mit
CATALOG
RUN usw.
auf dieses Volume wie üblich zugreifen.

er-Leser ·



Sonderangebot für Peeker-Leser

Wie schnell ist die MDB?

Die Datenübertragungsrate beträgt bei der MDB physikalisch 5 MBits/s. Dies ist ein theoretischer und niemals erreichbarer Wert. Auf Spur-Sektor- bzw. Blockebene beträgt die praktisch erzielbare Übertragungsrate ca. 44K/s bis 47,5K/s, je nach Betriebssystem. Einige anschauliche Beispiele: Auf Sektorebene läßt sich ein 140K-DOS-Volume mit dem Programm MDB.KOPY in 6,2s auf ein anderes 140K-Volume überspielen. Auf Dateiebene läßt sich eine 1-Megabyte-ProDOS-Datei mit dem Programm PROFID in 122s duplizieren. Dies sind umgerechnet 17.000 Bytes/s für Lesen bzw. Schreiben.

Was läuft mit der MDB?

Sie können normale Disk-II-Laufwerke und beispielsweise Erphi-160-Spur-Laufwerke im Mischbetrieb einsetzen. Die zusätzliche Verwendung von RAM-Karten ist bei der MDB überflüssig, da mit diesen in der Regel keine höheren Übertragungsraten erzielt werden können. Die Accelerator IIe funktioniert einwandfrei in Verbindung mit dem Megaboard-Controller (Slot-7-Schalter auf 1MHz einstellen). Die Speedemon funktioniert jedoch wegen des Cache-Speichers leider nicht.

DB-Meister

Speziell für die zukünftigen MDB-Besitzer wurde das Adreß- und Schemabriefprogramm DB-Meister, das in der normalen Diskettenversion DM 290,- kostet, zum Festplattenbetrieb umgeschrieben. Technische Daten:

– Datensatzlänge bis zu 230 Zeichen, aufteilbar in bis zu 25 Felder; maximal 1000 Datensätze pro Volume
– 2 variable Indexfelder und 1 zusätzliches, konstantes Suchfeld.
– Maskengenerator-Modul für die freie Definition von Bildschirmeingabe- und Ausdruck-Masken.
– Dateipflege-Modul zum Neueingeben, Ändern, Löschen, Ansehen, Bildschirm-Ausdruck usw.
– Sortier- und Filter-Module zum Sortieren und Untersortieren nach Indexfeldern und zum selektiven oder kumulierenden Filtern nach beliebigen Feldern.
– Druck-Modul zum Ausdrucken von Selbstklebe-Etiketten, tabellarischen Listen und Schemabriefen mit automatischem Einschub von Adressen und Anreden sowie weiteren Feldeinschüben im Briefkörper; Einzelblatt und Endlospapier verwendbar; Druckertreiber für Typenraddrucker mit anderer Typenradbelegung können eingebunden werden.
– Brief-Modul zum Schreiben der Schemabriefe; bei Bedarf separat verwendbar.
– Schnelles Backup-Programm (nur 10 Minuten für 2,5 Megabytes bzw. 18 DB-Meister-Dateien).
– Diverse Spezial-Module zur Indexfeld-Umdefinierung, zur Sortierung nach Kundennummern usw.

Alle Programm-Module laufen auf einem Apple IIe oder II+ mit MDB (Ausnahme: Das Brief-Modul funktioniert nur auf dem IIe mit 80-Zeichenkarte, doch können beliebige Fremdtexte eingelesen werden.).

Das DB-Meister-Programm eignet sich für normale Adreßverwaltung sowie für Dateien aller Art mit vielen, aber kleinen Datensätzen (max. 230 Zeichen). Das Programm ist nicht für das Schreiben von Rechnungen o.ä. gedacht.

Die Verarbeitungsgeschwindigkeit des „DB-Meisters" ist bei der MDB extrem schnell. Einige Sekundenangaben, bezogen auf eine Datei mit 500 Datensätzen zu je 220 Zeichen:

– Modul-Wechsel: 1s
– Suche nach Indexfelder: 0,3s
– Suche nach Nicht-Indexfeldern: 17s
– Filtern und Untersortieren: 25s
– Bildschirm-„Drucken": 38s

Noch ein etwas anschaulicheres Beispiel: Wenn Suchfelder neu vergeben werden sollen, muß die alte Indexdatei (ca. 50 Sektoren) und die ganze Hauptdatei (ca. 450 Sektoren) eingelesen werden und dann nach dem automatischen Heraussuchen der neuen Suchwörter die neue Indexdatei abgespeichert werden. Dieser normalerweise zeitraubende Prozeß dauert auf der MDB nur 14s.

Wie wird bestellt?

Sie senden Ihre Bestellung an den Hüthig-Software-Service. Sie erhalten dann von der Firma Frank & Britting eine Vorausrechnung, nach deren Überweisung Ihnen von dort die MDB10 bzw. MDB20, der Megaboard-Controller, das Handbuch und die Konfigurierungsdisketten mit 6 Monaten Garantie geliefert werden. Gleichzeitig erhalten Sie vom Hüthig-Software-Service das DB-Meister-Programm (2 Disketten und Handbuch) in der für die MDB bereits angepaßten Version. Nach einer geringfügigen Änderung im Hello-Programm können Sie diese Neuversion des DB-Meisters übrigens auch zusätzlich auf normalen 35-Spur-Laufwerken einsetzen.

Zur Bestellung können Sie eine der im Peeker eingehefteten Bestellkarten verwenden. Stichwort:

1 x MDB10-Sonderangebot für DM 2799,-
oder
1 x MDB20-Sonderangebot für DM 3199,-

Hüthig PUBLIKATION

SOFTWARE SERVICE
Im Weiher 10 · 6900 Heidelberg 1

Hold-Puffer) entsprechend gesetzt und der Original-Wert von DSPEND zurückgespeichert. Wenn wir uns gerade im Modus COMPARE befinden, muß das Display auf den neuen Stand gebracht werden, weil Hold-Puffer und Puffer jetzt übereinstimmen. Ansonsten endet HOLD gleich mit einem RTS zum Command Interpreter.

LIMIT (Z. 729-749) Wird sowohl von COMPARE als auch von MOVE benutzt: Zuerst wird zur Startadresse die Länge des Hold-Puffers addiert und das Ergebnis ggf. auf MAXBUF begrenzt. Danach folgt ein Vergleich mit der tatsächlich angegebenen (oder von GETPRM gesetzten) Endadresse. Wenn diese Endadresse größer als der vorher ermittelte Wert ist, so wird sie begrenzt. LIMIT liefert die Endadresse in A-Y.

MOVE (Z. 756-787) ist das Gegenstück zu HOLD, nur wird hier vorher HLSET geprüft und ggf. mit „No Hold Buffer set" abgebrochen. Nach dem Holen der Start/Endadresse über GETPRM wird durch LIMIT entweder auf Start+HLBUF-Länge oder auf MAXBUF begrenzt. Danach folgt eine Kopieraktion von HLBUF nach BUFFER. MOVE endet in jedem Fall mit einem REDUMP, das den neuen Puffer-Inhalt anzeigt.

3. ED.FUNCS2 und ED.CMDS

ED.FUNCS2 ist noch kürzer geworden als in der ersten Version von EDIT: Hier steht nur noch ein Sprung zu CMDERR für die Routine GETVAR, bevor es mit ED.CMDS weitergeht.
Bei ED.CMDS finden Sie die neuen Kommando-Zeichen nebst den dazugehörigen Startadressen. Die Definition der Startadresse von HLBUF (Z. 55) hat jetzt einen Sinn bekommen.

Hinweis:
Der dritte und letzte Teil dieser Serie erscheint in Heft 8/86.

Peeker-Sammeldisk # 19

DOS-3.3-Diskette; Heft 7/1986; Einzelpreis DM 28,–; Fortsetzungspreis DM 20,–

(1) = Zweck; (2) = Heft/Seitenzahl; (3) = Gerätekonfiguration;
(4) = Betriebssystem; (5) = Programmstart;
(6) = Sonstiges

A 002 STIEHL
A 003 HELLO
B 006 DDMOVER
B 050 REGISTER.UTILITIES
B 024 REGISTER.RUNTIME
B 009 REGISTERSTARTER.OBJ
B 030 REGISTER-REDIGIERER.OBJ
B 018 REGISTER-SORTIERER.OBJ
B 019 REGISTERMISCHER.OBJ
B 020 REGISTER-DRUCKER.OBJ
B 018 REGISTER-UMWANDLER.OBJ
A 007 REGISTERTESTER
A 008 REGISTERUMDREHER
A 007 REGISTERTEILER

(1) Programmpaket zum Erstellen, Sortieren, Ausdrucken und Pflegen von Buchregistern (= Seitenregistern) und (zweisprachigen) Glossaren; (2) Heft 7/86, S. ??; (3) Apple II+/e/c mit 2 Diskettenlaufwerken (Disk II oder Erphi-160-Spur-Drive); auch 256K-RAM-Disk oder MDB-Festplatte verwendbar; beliebiger Slot, doch müssen beide Drives an demselben Slot angeschlossen sein; (4) Diversi-DOS 2C oder ersatzweise mit Einschränkungen DOS-3.3; (5) RUN STIEHL von Drive 1; zuvor formatierte Leerdiskette in Drive 2 einlegen.

A 005 MOUSORY
A 013 MOUSORY.DEMO
A 019 MOUSORY.GAME
B 002 MOUSORY.BELL
B 024 MOUSORY.SET1
B 022 MOUSORY.SET2
B 026 MOUSORY.SET3
B 034 MOUSORY.BILD

(1) Memory-Spiel mit Hires-Spielkarten; (2) Heft 7/86, S. ??; (3) Apple II+/e/c; Apple-Maus optional, d.h. auch Cursortasten verwendbar; (4) DOS 3.3 oder ProDOS; (5) RUN MOUSORY

A 002 START
A 005 HP
A 003 UP1
A 003 UP2
T 007 T.RUN.FILE
B 002 RUN.FILE

(1) Demo-Programm für Chain-Routine (= Übergabe von Variablen an das nächste Programm-Modul) unter Applesoft; (2) Heft 7/86, S. ??; (3) Apple II+/e/c; (4) DOS 3.3; aus verschiedenen Gründen (HIMEM u. a.) nicht lauffähig unter ProDOS!; (5) RUN START

A 015 CLOCK
T 020 CLOCK.EXEC

(1) Anpassungsprogramm für Hardware-Uhr TIME II; (2) Heft 7/86, S. ??; (3) Apple II+/e mit TIME-II-Uhrenkarte; (4) ProDOS (nur unter Version 1.0.1 getestet); (5) RUN CLOCK oder vorher Original-STARTUP laden und dann EXEC CLOCK.EXEC; (6) Programme müssen zuvor von der Sammeldisk mit CONVERT oder DOSTOPRO auf Ihre ProDOS-Arbeitsdiskette konvertiert werden; Programme nur starten, wenn TIME II im Rechner installiert ist!

A 006 PRODOS.LIB.DEMO
T 074 T.PRODOS.LIB
B 009 PRODOS.LIB

(1) Muster-ProDOS-Runtime-Library für Stand-alone-Programme (Assembler, Applesoft oder Kyan-Pascal), die aus speicher- oder programmtechnischen Gründen auf das BASIC.SYSTEM verzichten müssen; (2) Heft 7/86, S. ??; (3) Apple II+/e/c; (4) ProDOS (jede Version); (5) RUN PRODOS. LIB.DEMO; (6) Programme müssen zuvor von der Sammeldisk mit CONVERT oder DOSTOPRO auf Ihre ProDOS-Arbeitsdiskette konvertiert werden.

Hüthig Software Service
Postfach 10 28 69 · 6900 Heidelberg

ED.FUNCS1.TXT

```
10F6:                PAGE
10F6:                ***************
10F6:                *---MONITOR---*
10F6:                ***************
10F6:                *
10F6:68              XMON:  PLA            ;die Returnadresse
10F7:8D 38 11                STA XMONRTS   ;innerhalb von EDIT
10FA:68                      PLA           ;wird gespeichert
10FB:8D 39 11                STA XMONRTS+1
10FE:                *
10FE:A2 00                   LDX #00       ;SAVE von CMDSTR,
1100:BD 00 02        SCMDSTR: LDA CMDSTR,X ;der Monitor benutzt
1103:9D 00 26                STA EDITEND,X ;ebenfalls $200... als
1106:E8                      INX           ;Input-Buffer!
1107:D0 F7                   BNE SCMDSTR
1109:                *
1109:20 5B 0B                JSR DOCR      ;ein <CR>
110C:A9 1C                   LDA #>MONRTS  ;Rücksprungadresse
110E:8D F9 03                STA YVEC      ;zu EDIT in den
1111:A9 11                   LDA #<MONRTS  ;Ctrl-Y Vektor
1113:8D FA 03                STA YVEC+1
1116:20 88 0D                JSR DOSCONN   ;DOS anhängen
1119:4C 69 FF                JMP MONITOR   ;"RTS": Ctrl-Y
111C:                *
111C:20 7D 0D        MONRTS: JSR DISCONN   ;DOS wieder ab
111F:A2 00                   LDX #00       ;RESTORE CMDSTR
1121:BD 00 26        RCMDSTR: LDA EDITEND,X
1124:9D 00 02                STA CMDSTR,X  ;von EDITEND nach
1127:E8                      INX           ;$200...
1128:D0 F7                   BNE RCMDSTR
112A:                *
112A:AD 39 11                LDA XMONRTS+1 ;RESTORE der
112D:48                      PHA           ;RTS-Adresse
112E:AD 38 11                LDA XMONRTS   ;innerhalb von EDIT
1131:48                      PHA
1132:4C 3A 11                JMP REPRINT   ;RESTORE Bildschirm
1135:                *
1135:4C AA 0A        MONERR: JMP CMDERR
1138:                XMONRTS: DS 2         ;RTS (intern)
113A:                *
113A:20 BD 08        REPRINT: JSR TITLE    ;Neudruck des
113D:20 75 10                JSR PRINFO    ;Bildschirms
1140:AD AE 0C                LDA BPLINE    ;Bytes per Line
1143:20 7F 13                JSR SETDUMP   ;DUMP BUFFER
1146:20 47 0B                JSR SETV221   ;VTAB22, HTAB 1
1149:20 2D 09                JSR RTYP2     ;CMDSTR (alt)
114C:4C 1C 09                JMP RETYPE    ;CMDSTR (neu)
114F:                *
114F:                ************
114F:                *---QUIT---*
114F:                ************
114F:                *
114F:20 4F 0B        QUIT:  JSR SETV22    ;VTAB 22
1152:20 9C 0B                JSR CLEOP
1155:4C 98 0D                JMP EXIT      ;DOS Kaltstart
1158:                *
1158:                *****************
1158:                *-HALFUP/LINEUP-*
1158:                *****************
1158:                *
1158:A9 BC           HALFUP: LDA #'<'      ;halbe Seite auf
115A:20 EB 11                JSR MORE      ;"<<<<<"?
115D:20 FE 11                JSR MULT8     ;"<" = 8 * ","
1160:4C 6A 11                JMP LINUPZ
1163:                *
1163:A9 AC           LINEUP: LDA #','      ;Zeile aufwärts
1165:20 EB 11                JSR MORE      ;",,,,"?
1168:A0 00                   LDY #00
116A:20 04 12        LINUPZ: JSR XNBYTES   ;mal Bytes/Zeile
116D:38                      SEC
116E:AD B2 0C                LDA DSPSTART  ;alter Start
1171:ED 31 12                SBC SCLARG    ;gerade berechnet
1174:A8                      TAY
1175:AD B3 0C                LDA DSPSTART+1
1178:ED 32 12                SBC SCLARG+1
117B:B0 03           LINZRO? BCS TSTBTM    ;Start noch > 000
117D:A9 00           SETZRO: LDA #00       ;sonst Korrektur
117F:A8                      TAY           ;auf 00 00
```

```
1180:CD B1 0C        TSTBTM: CMP DSPBTM+1  ;SADDR < DSPBTM?
1183:90 07                   BCC SETBTM    ;ja
1185:D0 4B                   BNE DOSCRL
1187:CC B0 0C                CPY DSPBTM
118A:B0 46                   BCS DOSCRL    ;nein
118C:AD B1 0C        SETBTM: LDA DSPBTM+1  ;sonst Korrektur
118F:AC B0 0C                LDY DSPBTM    ;auf DSPBTM
1192:90 3E                   BCC DOSCRL    ;"always"
1194:                *
1194:                *****************
1194:                *-HALFDN/LINEDN-*
1194:                *****************
1194:                *
1194:A9 BE           HALFDN: LDA #'>'      ;halbe Seite ab
1196:20 EB 11                JSR MORE      ;">>>>"?
1199:20 FE 11                JSR MULT8     ;">" = 8 * "."
119C:4C A6 11                JMP LINDNZ
119F:                *
119F:A9 AE           LINEDN: LDA #'.'      ;Zeile ab
11A1:20 EB 11                JSR MORE      ;"...."?
11A4:A0 00                   LDY #00
11A6:20 04 12        LINDNZ: JSR XNBYTES   ;mal Bytes/Zeile
11A9:18                      CLC
11AA:AD B2 0C                LDA DSPSTART  ;alter Start
11AD:6D 31 12                ADC SCLARG
11B0:A8                      TAY
11B1:AD B3 0C                LDA DSPSTART+1
11B4:6D 32 12                ADC SCLARG+1
11B7:CD B9 0C                CMP DSPTOP+1  ;> Display-Ende?
11BA:90 16                   BCC DOSCRL
11BC:D0 05                   BNE LINMAX
11BE:CC B8 0C                CPY DSPTOP
11C1:90 0F                   BCC DOSCRL    ;ist kleiner
11C3:AD B8 0C        LINMAX: LDA DSPTOP    ;sonst Limit
11C6:ED AF 0C                SBC NBYTES    ;auf letzte Zeile
11C9:A8                      TAY
11CA:AD B9 0C                LDA DSPTOP+1
11CD:E9 00                   SBC #00
11CF:4C 7B 11                JMP LINZRO    ;Test, ob < DSPSTART
11D2:                *
11D2:8C B2 0C        DOSCRL: STY DSPSTART  ;neuer Start
11D5:8D B3 0C                STA DSPSTART+1
11D8:AD B9 0C                LDA DSPTOP+1  ;DSPTOP => DSPEND,
11DB:AC B8 0C                LDY DSPTOP    ;d.h. eine volle
11DE:8D B7 0C                STA DSPEND+1  ;Schirmseite, wenn
11E1:8C B6 0C                STY DSPEND    ;möglich
11E4:20 A2 13                JSR DUMP
11E7:20 6F 0B                JSR CLRDS2    ;löscht den Rest des
11EA:60              SCRLDONE: RTS         ;Display-Feldes
11EB:                *
11EB:AE 9E 0C        MORE:  LDX CMDIDX    ;Test auf mehrere
11EE:A0 01                  LDY #01       ;gleiche Zeichen
11F0:DD 00 02        MR1:   CMP CMDSTR,X  ;in CMDSTR
11F3:D0 04                  BNE NOMORE    ;Ende...
11F5:C8                     INY
11F6:E8                     INX
11F7:D0 F7                  BNE MR1
11F9:8E 9E 0C        NOMORE: STX CMDIDX
11FC:98                     TYA           ;Zahl gleicher Zeichen
11FD:60                     RTS
11FE:                *
11FE:A2 08           MULT8: LDX #08
1200:A0 00                  LDY #00       ;Hi-Byte
1202:F0 03                  BEQ MULTZ
1204:                *
1204:AE AF 0C        XNBYTES: LDX NBYTES   ;Faktor
1207:                *
1207:8D 33 12        MULTZ: STA MFAC       ;Low-Byte
120A:8C 34 12               STY MFAC+1
120D:A9 00                  LDA #00
120F:8D 31 12                STA SCLARG    ;Ergebnis
1212:8D 32 12        MLOOP: STA SCLARG+1
1215:CA                     DEX           ;etwas primitiv,
1216:30 12                  BMI MULDONE   ;funktioniert
1218:18                     CLC           ;aber für X=0!
1219:AD 31 12                LDA SCLARG
121C:6D 33 12                ADC MFAC
121F:8D 31 12                STA SCLARG
1222:AD 32 12                LDA SCLARG+1
1225:6D 34 12                ADC MFAC+1
1228:90 E8                   BCC MLOOP     ;"always?"
122A:AD 31 12        MULDONE: LDA SCLARG
122D:AC 32 12                LDY SCLARG+1
1230:60                      RTS
```

```
1231:            169 *
1231:            170 SCLARG: DS 2          ;Scroll-Argument
1233:            171 MFAC:  DS  2          ;MULTZ-Faktor
1235:            172 *
1235:A0 04       175 TSTMODE: LDY #VMODE-NXTMODE ;Test, ob
1237:A9 00       176         LDA #00       ;irgend ein Modus an
1239:19 A8 0C    177 TSTMD: ORA NXTMODE,Y
123C:30 03       178         BMI MODSET    ;ja
123E:88          179         DEY           ;wenn nicht, dann
123F:D0 F8       180         BNE TSTMD     ;Exit mit "EQ", sonst
1241:60          181 MODSET: RTS           ;mit "MI"
1242:            182 *
1242:A9 00       183 NOMODE: LDA #00       ;alle anderen Modes
1244:AA          184         TAX           ;vor dem Setzen eines
1245:9D A9 0C    185 MDSOFF: STA AMODE,X   ;neuen Modus OFF
1248:E8          186         INX
1249:E0 04       187         CPX #VMODE-NXTMODE
124B:90 F8       188         BCC MDSOFF
124D:60          189         RTS
124E:            190 *
124E:AD B7 0C    192 SAVEND: LDA DSPEND+1 ;Zwischenspeichern von
1251:AC B6 0C    193         LDY DSPEND    ;DSPEND für REDUMP und
1254:8D 6F 12    194         STA ENDSAV+1  ;HOLD.
1257:8C 6E 12    195         STY ENDSAV
125A:AD A3 0C    196         LDA EADDR+1   ;HOLD: DSPEND=EADDR
125D:AC A2 0C    197         LDY EADDR     ;REDUMP: Limit DSPTOP
1260:60          198         RTS
1261:            199 *
1261:AD 6F 12    200 RESEND: LDA ENDSAV+1 ;Restore von DSPEND
1264:AC 6E 12    201         LDY ENDSAV    ;nach REDUMP/HOLD
1267:8D B7 0C    202         STA DSPEND+1
126A:8C B6 0C    203         STY DSPEND
126D:60          204         RTS
126E:            205 *
126E:            206 ENDSAV: DS 2          ;Temp für DSPEND
1270:            207 *
1270:AD A1 0C    208 SADSTART: LDA SADDR+1 ;SADDR => DSPSTART
1273:AC A0 0C    209         LDY SADDR
1276:8D B3 0C    210         STA DSPSTART+1
1279:8C B2 0C    211         STY DSPSTART
127C:60          212         RTS
127D:            213 *
127D:20 35 12    215 REDUMP: JSR TSTMODE  ;irgendein Modus an?
1280:10 05       216         BPL NODMP     ;nein
1282:2C 9C 0C    217         BIT NODSP     ;Display OFF?
1285:10 03       218         BPL REDMP1    ;nein, ist ON
1287:4C 35 13    219 NODMP: JMP REDONE     ;RTS
128A:A9 05       220 REDMP1: LDA #05       ;VTAB 5
128C:20 55 0B    221         JSR SETVTB
128F:A9 10       222         LDA #16
1291:8D FF 14    223         STA DLINES    ;Zeilen-Zähler
1294:AD A2 0C    224         LDA EADDR     ;EADDR kleiner/gleich
1297:18          225         CLC           ;(!)
1298:ED B2 0C    226         SBC DSPSTART  ;DSPSTART?
129B:AD A3 0C    227         LDA EADDR+1
129E:ED B3 0C    228         SBC DSPSTART+1
12A1:90 E4       229         BCC NODMP     ;ja, kein REDUMP
12A3:AD B6 0C    230         LDA DSPEND    ;SADDR größer/gleich
12A6:18          231         CLC           ;(!)
12A7:ED A0 0C    232         SBC SADDR     ;DSPEND?
12AA:AD B7 0C    233         LDA DSPEND+1
12AD:ED A1 0C    234         SBC SADDR+1
12B0:90 D5       235         BCC NODMP     ;ja, kein REDUMP
12B2:AD A0 0C    236         LDA SADDR     ;SADDR auf
12B5:2D FC 14    237         AND CRMASK    ;"Zeilenanfang"
12B8:4D A0 0C    238         EOR SADDR     ;=> NYBTES maskiert
12BB:8D B4 0C    239         STA NXTADDR
12BE:ED B2 0C    240         SBC DSPSTART  ;wenn SADDR kleiner
12C1:A8          241         TAY           ;oder gleich DSPSTART,
12C2:AD A1 0C    242         LDA SADDR+1   ;dann REDUMP ab SADDR
12C5:8D B5 0C    243         STA NXTADDR+1
12C8:ED B3 0C    244         SBC DSPSTART+1
12CB:8D 94 0C    245         STA TEMP      ;Differenz hi
12CE:90 03       246         BCC REDSP     ;DSPSTART > SADDR
12D0:98          247         TYA
12D1:D0 0F       248         BNE NOTLIN0   ;SADDR > DSPSTART
12D3:AD B3 0C    249 REDSP: LDA DSPSTART+1
12D6:8D B5 0C    250         STA NXTADDR+1 ;DSPSTART bleibt
12D9:AC B2 0C    251         LDY DSPSTART  ;Startadresse
12DC:8C B4 0C    252         STY NXTADDR
12DF:4C 00 13    253         JMP DODUMP
12E2:            254 *
12E2:A0 05       255 NOTLIN0: LDY #5       ;Start mit VTAB6
12E4:C8          256 SETDLIN: INY
12E5:CE FF 14    257         DEC DLINES    ;Bildschirm zu Ende?
12E8:F0 4B       258         BEQ REDONE    ;ja, Exit
12EA:38          259         SEC
12EB:ED AF 0C    260         SBC NBYTES    ;Differenz lo in Acc
12EE:D0 05       261         BNE STNEXT
12F0:0D 94 0C    262         ORA TEMP      ;genau auf 00?
12F3:F0 07       263         BEQ SETDLZ    ;ja
12F5:B0 ED       264 STNEXT: BCS SETDLIN
12F7:CE 94 0C    265         DEC TEMP      ;Differenz Hi
12FA:10 E8       266         BPL SETDLIN
12FC:98          267 SETDLZ: TYA           ;ergibt Zeilennummer:
12FD:20 55 0B    268         JSR SETVTB    ;(Diff/NBYTES + 5)
1300:AD A2 0C    269 DODUMP: LDA EADDR     ;EADDR auf Zeilenende
1303:0D FC 14    270         ORA CRMASK
1306:18          271         CLC
1307:69 01       272         ADC #01
1309:8D A2 0C    273         STA EADDR
130C:90 03       274         BCC DODMP2
130E:EE A3 0C    275         INC EADDR+1
1311:20 4E 12    276 DODMP2: JSR SAVEND    ;Speichert DSPEND und
1314:CD B7 0C    277         CMP DSPEND+1  ;holt EADDR:
1317:90 0D       278         BCC DODMPZ    ;EADDR > DSPEND?
1319:D0 05       279         BNE SETEND    ;ja
131B:CC B6 0C    280         CPY DSPEND
131E:90 06       281         BCC DODMPZ    ;nein
1320:AD B7 0C    282 SETEND: LDA DSPEND+1 ;sonst Limit auf
1323:AC B6 0C    283         LDY DSPEND    ;DSPEND
1326:8D B7 0C    284 DODMPZ: STA DSPEND+1
1329:8C B6 0C    285         STY DSPEND
132C:20 9E 16    286         JSR SCPIDX    ;setzt COMPARE-Index
132F:20 CB 13    287         JSR DLOOP     ;DUMP
1332:20 61 12    288         JSR RESEND    ;Restore DSPEND
1335:60          289 REDONE: RTS
1336:            290 *
1336:            292 ***************************
1336:            293 *---ASCII/HEX/VBM-DUMP---*
1336:            294 ***************************
1336:            295 *
1336:A0 00       296 DUMPA: LDY #AMODE-AMODE ;Index zu AMODE
1338:A9 20       297         LDA #$20      ;$20 Bytes per Zeile
133A:D0 0A       298         BNE SETMOD1
133C:A0 01       299 DUMPH: LDY #HMODE-AMODE ;Index zu HMODE
133E:A9 10       300         LDA #$10      ;$10 Bytes per Zeile
1340:D0 04       301         BNE SETMOD1
1342:A0 03       302 DUMPV: LDY #VMODE-AMODE ;Index zu VMODE
1344:A9 08       303         LDA #$08      ;$08 Bytes per Zeile
1346:            304 *
1346:8D AD 0C    305 SETMOD1: STA LBYTES   ;Bytes per Zeile
1349:8C A8 0C    306         STY NXTMODE   ;nächster Modus
134C:20 C5 0B    307         JSR GETPRM    ;holt SADDR/EADDR
134F:A9 00       308         LDA #00       ;aus CMDSTR
1351:8D B1 0C    309         STA DSPBTM+1  ;Untergrenze für
1354:8D B0 0C    310         STA DSPBTM    ;DUMP: $00 00
1357:8D B8 0C    311         STA DSPTOP    ;Obergrenze: MAXBUF
135A:A9 02       312         LDA #MAXBUF
135C:8D B9 0C    313         STA DSPTOP+1
135F:20 70 12    314         JSR SADSTART  ;SADDR => DSPSTART
1362:            315 *
1362:AD A3 0C    316 SETMOD2: LDA EADDR+1 ;EADDR => DSPEND
1365:AC A2 0C    317         LDY EADDR
1368:8D B7 0C    318         STA DSPEND+1
136B:8C B6 0C    319         STY DSPEND
136E:20 42 12    320         JSR NOMODE    ;alle Modes aus
1371:A9 80       321         LDA #$80
1373:AC A8 0C    322         LDY NXTMODE   ;und neuer Modus an
1376:99 A9 0C    323         STA AMODE,Y
1379:20 63 0B    324         JSR CLRDSP    ;Display-Feld clear
137C:AD AD 0C    325         LDA LBYTES    ;Bytes per Zeile
137F:            326 *
137F:8D AE 0C    327 SETDUMP: STA BPLINE
1382:2C A9 0C    328         BIT AMODE     ;für ASCII-Dump:
1385:30 06       329         BMI NOT40     ;immer $20 Bytes
1387:2C 9A 0C    330         BIT IS40      ;40 Zeichen?
138A:10 01       331         BPL NOT40     ;für 40Z außer ASCII:
138C:4A          332         LSR A         ;Bytes per Zeile / 2
138D:8D AF 0C    333 NOT40: STA NBYTES
1390:48          334         PHA
1391:38          335         SEC
1392:E9 01       336         SBC #01       ;00x00000 => 000yyyyy
1394:8D FC 14    337         STA CRMASK    ;Maske für Zeilenende
1397:68          338         PLA
1398:4A          339         LSR A         ;Zwischenspalten-Index
1399:0D 9A 0C    340         ORA IS40      ;nur für Hexdump bei
139C:0D AB 0C    341         ORA CMODE     ;80 Zeichen, sonst
139F:8D FD 14    342 STHALF: STA CRHALF    ;auf $8x gesetzt
13A2:            343 *
13A2:2C 9C 0C    345 DUMP:  BIT NODSP      ;Display OFF?
13A5:30 21       346         BMI DMPEXIT   ;ja
13A7:A9 05       347         LDA #5        ;VTAB 5
13A9:20 55 0B    348         JSR SETVTB
```

```
13AC:A9 1Ø      349          LDA #16        ;Bildschirm: 16 Zeilen
13AE:ØD 9B ØC   35Ø          ORA PRINTER    ;Printer:
13B1:8D FF 14   351          STA DLINES     ;keine Begrenzung ($FF)
13B4:AD B3 ØC   352          LDA DSPSTART+1
13B7:AC B2 ØC   353          LDY DSPSTART   ;DUMP-Start ab DSPSTART
13BA:8D B5 ØC   354          STA NXTADDR+1
13BD:8C B4 ØC   355          STY NXTADDR
13CØ:2Ø 9E 16   356          JSR SCPIDX     ;HLBUF-Index (COMPARE)
13C3:2Ø 35 12   357          JSR TSTMODE    ;irgend ein Modus an?
13C6:3Ø Ø3      358          BMI DLOOP
13C8:4C B4 14   359 DMPEXIT: JMP DUMPQT     ;nein, RTS
13CB:           36Ø *
13CB:AD B5 ØC   361 DLOOP: LDA NXTADDR+1 ;wird im Loop erhöht
13CE:AC B4 ØC   362          LDY NXTADDR
13D1:2Ø Ø9 ØB   363          JSR PRADDR     ;"xxx- "
13D4:2Ø B5 14   364          JSR GNLINE     ;eine Zeile aus BUFFER
13D7:A2 ØØ      365          LDX #ØØ        ;nach LBUF, X=Index
13D9:2C A9 ØC   366          BIT AMODE      ;nur ASCII?
13DC:1Ø Ø3      367          BPL TSTVMOD
13DE:4C 6B 14   368          JMP DMPASC     ;"Verlängerung"...
13E1:2C AC ØC   369 TSTVMOD: BIT VMODE      ;VBM-Dump?
13E4:1Ø Ø3      37Ø          BPL DMPHEX
13E6:4C 84 14   371          JMP DMPVBM
13E9:           372 *
13E9:BD Ø2 15   373 DMPHEX: LDA LBUF,X
13EC:2Ø 1D ØB   374          JSR PRBYTE     ;druckt Byte in Hex
13EF:A9 AØ      375          LDA #$AØ
13F1:2Ø ED FD   376          JSR COUT       ;ein <Space> danach
13F4:E8         377          INX
13F5:EC FD 14   378          CPX CRHALF     ;Zwischenspalte?
13F8:DØ Ø3      379          BNE DMPHX2
13FA:2Ø ED FD   38Ø          JSR COUT       ;noch ein <Space>
13FD:EC Ø1 15   381 DMPHX2: CPX LTOP        ;Zeilenende?
14ØØ:9Ø E7      382          BCC DMPHEX
14Ø2:2Ø B2 ØB   383          JSR CLEOL      ;Clear End_of_Line
14Ø5:A2 ØØ      384          LDX #ØØ
14Ø7:2C AB ØC   385          BIT CMODE      ;COMPARE?
14ØA:3Ø ØF      386          BMI DMPCMP
14ØC:A9 39      387          LDA #57        ;8ØZ: HTAB58 für ASCII
14ØE:2C 9A ØC   388          BIT IS4Ø
1411:1Ø Ø2       389          BPL HEXDONE
1413:A9 1E      39Ø          LDA #3Ø        ;4ØZ: HTAB31 für ASCII
1415:2Ø 89 ØB   391 HEXDONE: JSR SETCH
1418:4C 6B 14   392          JMP DMPASC
141B:           393 *
141B:A9 1D      394 DMPCMP: LDA #29        ;8ØZ:HTAB 3Ø für "***"
141D:2C 9A ØC   395          BIT IS4Ø
142Ø:1Ø Ø2      396          BPL CMP1
1422:A9 11      397          LDA #17        ;4ØZ:HTAB 18 für "***"
1424:2Ø 89 ØB   398 CMP1:  JSR SETCH
1427:A9 AA      399          LDA #'*'
1429:2Ø ED FD   4ØØ          JSR COUT       ;drei Sternchen
142C:2Ø ED FD   4Ø1          JSR COUT
142F:2Ø ED FD   4Ø2          JSR COUT
1432:A9 AØ      4Ø3          LDA #$AØ       ;und ein <Space>
1434:2Ø ED FD   4Ø4          JSR COUT
1437:2Ø 22 15   4Ø5 CMP2:  JSR GCBYTE     ;ein Byte aus HLBUF
143A:DD Ø2 15   4Ø6          CMP LBUF,X
143D:DØ ØB      4Ø7          BNE CMP3       ;ungleich!
143F:A9 AE      4Ø8          LDA #'.'       ;gleich: es wird
1441:2Ø ED FD   4Ø9          JSR COUT       ;"..." gedruckt
1444:2Ø ED FD   41Ø          JSR COUT
1447:4C 52 14   411          JMP CMP4
144A:2Ø 1D ØB   412 CMP3:  JSR PRBYTE     ;mit ungleichem Byte
144D:A9 AD      413          LDA #'-'
144F:8D BA ØC   414          STA SUCCEED    ;SUCCEED auf "-"
1452:A9 AØ      415 CMP4:  LDA #$AØ       ;<Space> nach
1454:2Ø ED FD   416          JSR COUT       ;"..." oder Hexbyte
1457:E8         417          INX
1458:EC Ø1 15   418          CPX LTOP       ;Zeilenende?
145B:9Ø DA      419          BCC CMP2
145D:A2 ØØ      42Ø          LDX #ØØ        ;Index in LBUF zurück
145F:A9 3C      421          LDA #6Ø        ;8ØZ: HTAB61 für ASCII
1461:2C 9A ØC   422          BIT IS4Ø
1464:1Ø Ø2      423          BPL CMPDONE
1466:A9 23      424          LDA #35        ;4ØZ: HTAB36 für ASCII
1468:2Ø 89 ØB   425 CMPDONE: JSR SETCH
146B:           426 *
146B:BD Ø2 15   427 DMPASC: LDA LBUF,X
146E:29 7F      428          AND #$7F
147Ø:C9 2Ø      429          CMP #$2Ø       ;druckbar?
1472:BD Ø2 15   43Ø          LDA LBUF,X
1475:BØ Ø2      431          BCS DMPAC2     ;Carry vom Vergleich
1477:A9 AE      432          LDA #'.'       ;Controls => "."
1479:2Ø E6 ØA   433 DMPAC2: JSR PRCHAR
147C:E8         434          INX
147D:EC Ø1 15   435          CPX LTOP       ;Zeilenende?
```

```
148Ø:9Ø E9      436          BCC DMPASC
1482:BØ 1D      437          BCS LINDON
1484:           438 *
1484:BD Ø2 15   439 DMPVBM: LDA LBUF,X
1487:2Ø 33 ØB   44Ø          JSR PRBITS     ;Byte als Bits drucken
148A:E8         441          INX
148B:EC Ø1 15   442          CPX LTOP       ;Zeilende?
148E:BØ Ø8      443          BCS VLNDON     ;Zeilenende!
149Ø:A9 AØ      444          LDA #$AØ       ;sonst ein <Space>
1492:2Ø ED FD   445          JSR COUT
1495:4C 84 14   446          JMP DMPVBM
1498:2C 9A ØC   447 VLNDON: BIT IS4Ø       ;bei 4ØZ und 4 Bytes
149B:1Ø Ø4      448          BPL LINDON     ;geht DMPVBM über das
149D:A5 24      449          LDA CH         ;Zeilenende hinaus =>
149F:FØ Ø6      45Ø          BEQ LINDN1     ;auf nächster Zeile
14A1:           451 *
14A1:2Ø B2 ØB   452 LINDON: JSR CLEOL      ;Clear End_of_Line
14A4:2Ø 5B ØB   453          JSR DOCR       ;<CR>
14A7:CE FF 14   454 LINDN1: DEC DLINES     ;Zeilen-Zähler
14AA:FØ Ø8      455          BEQ DUMPQT
14AC:2C ØØ 15   456          BIT DDONE      ;DSPEND erreicht?
14AF:3Ø Ø3      457          BMI DUMPQT
14B1:4C CB 13   458          JMP DLOOP      ;nächste Zeile
14B4:6Ø         459 DUMPQT: RTS
14B5:           46Ø *
14B5:A2 ØØ      461 GNLINE: LDX #ØØ        ;holt eine Dump-Zeile
14B7:2Ø CC 14   462 GNLINZ: JSR GNBYTE     ;aus BUFFER
14BA:9D Ø2 15   463          STA LBUF,X
14BD:E8         464          INX
14BE:AD FE 14   465          LDA CRLF       ;Zeilenende?
14C1:FØ Ø5      466          BEQ GOTNLINE
14C3:2C ØØ 15   467          BIT DDONE      ;DSPEND erreicht?
14C6:1Ø EF      468          BPL GNLINZ
14C8:8E Ø1 15   469 GOTNLINE: STX LTOP     ;End-Index in LBUF
14CB:6Ø         47Ø          RTS
14CC:           471 *
14CC:AC B4 ØC   473 GNBYTE: LDY NXTADDR    ;Startadresse low
14CF:AD B5 ØC   474          LDA NXTADDR+1
14D2:18         475          CLC
14D3:69 2Ø      476          ADC #<BUFFER
14D5:8D DA 14   477          STA GNB1+2
14D8:B9 ØØ 2Ø   478 GNB1:  LDA BUFFER,Y    ;nächstes Byte
14DB:48         479          PHA
14DC:C8         48Ø          INY
14DD:DØ Ø3      481          BNE GNB2
14DF:EE B5 ØC   482          INC NXTADDR+1
14E2:8C B4 ØC   483 GNB2:  STY NXTADDR
14E5:98         484          TYA            ;Zeilenende nach
14E6:2D FC 14   485          AND CRMASK     ;diesem Byte?
14E9:8D FE 14   486          STA CRLF       ;ØØ, wenn ja
14EC:CC B6 ØC   487          CPY DSPEND
14EF:9Ø Ø6      488          BCC GNB3       ;noch kleiner DSPEND
14F1:AD B5 ØC   489          LDA NXTADDR+1
14F4:CD B7 ØC   49Ø          CMP DSPEND+1
14F7:6E ØØ 15   491 GNB3:  ROR DDONE       ;$8Ø, wenn = DSPEND
14FA:68         492          PLA
14FB:6Ø         493          RTS
14FC:           494 *
14FC:           495 CRMASK: DS 1           ;Maske für Zeilenende
14FD:           496 CRHALF: DS 1           ;Index für Zw.spalte
14FE:           497 CRLF:  DS  1           ;Flag: Zeilenende
14FF:           498 *
14FF:           499 DLINES: DS 1           ;Zeilenzahl für DUMP
15ØØ:           5ØØ DDONE: DS  1           ;Flag: Dump beendet
15Ø1:           5Ø1 LTOP:  DS  1           ;Bytes in LBUF
15Ø2:           5Ø2 LBUF:  DS  $2Ø         ;Zeilen-Buffer
1522:           5Ø3 *
1522:AC BD ØC   5Ø4 GCBYTE: LDY CPIDX      ;ein Byte aus HLBUF,
1525:AD BE ØC   5Ø5          LDA CPIDX+1    ;ein Test auf Ende ist
1528:18         5Ø6          CLC            ;hier nicht notwendig,
1529:69 22      5Ø7          ADC #<HLBUF    ;DDONE und LTOP sind
152B:8D 38 15   5Ø8          STA GCB1+2     ;von GNLINE gesetzt!
152E:EE BD ØC   5Ø9          INC CPIDX
1531:DØ Ø3      51Ø          BNE GCB1
1533:EE BE ØC   511          INC CPIDX+1
1536:B9 ØØ 22   512 GCB1:  LDA HLBUF,Y
1539:6Ø         513          RTS
153A:           515 *
153A:           516 ***********
153A:           517 *---SET---*
153A:           518 ***********
153A:           519 *
153A:2Ø C5 ØB   52Ø SET:   JSR GETPRM
153D:AD C3 ØC   521          LDA SETAD+1    ;"Default": interne
154Ø:AC C2 ØC   522          LDY SETAD      ;Startadresse (SETAD)
1543:2C A4 ØC   523          BIT GOTPRM1
1546:1Ø Ø6      524          BPL STSTART
```

```
1548:AD A1 ØC  525          LDA SADDR+1    ;Start spezifiziert:
154B:AC AØ ØC  526          LDY SADDR      ;SADDR => SETAD
154E:8D C3 ØC  527 STSTART: STA SETAD+1
1551:8C C2 ØC  528          STY SETAD
1554:8D A1 ØC  529          STA SADDR+1    ;für REDUMP
1557:8C AØ ØC  53Ø          STY SADDR
155A:18        531          CLC
155B:69 2Ø     532          ADC #<BUFFER
155D:8D ØA 16  533          STA SBYT1+2    ;Zieladresse
156Ø:AE 9E ØC  534          LDX CMDIDX
1563:BD ØØ Ø2  535          LDA CMDSTR,X
1566:2C A5 ØC  536          BIT GOTPRM2    ;EADDR spezifiziert?
1569:1Ø 11     537          BPL STHEX
156B:C9 AC     538          CMP #','       ;muß dann Komma sein
156D:DØ 58     539          BNE SETERR
156F:E8        54Ø          INX
157Ø:2Ø DA 15  541          JSR GETHEX     ;Parm3: <Byte>
1573:BØ 52     542          BCS SETERR     ;fehlt
1575:2Ø F8 15  543 FILL:    JSR SBYTE      ;von SADDR..EADDR
1578:9Ø FB     544          BCC FILL       ;wird mit <Byte>
157A:BØ 4E     545          BCS SETQT      ;gefüllt
157C:          546 *
157C:C9 BA     547 STHEX:   CMP #':'       ;danach Hexzahlen?
157E:DØ 1Ø     548          BNE SETASC
158Ø:E8        549          INX
1581:2Ø DA 15  55Ø SHEXZ:   JSR GETHEX
1584:BØ 44     551          BCS SETQT      ;Hexzahlen zuende
1586:2Ø F8 15  552          JSR SBYTE
1589:9Ø F6     553          BCC SHEXZ
158B:CE 9E ØC  554          DEC CMDIDX     ;für Fehleranzeige
158E:BØ 2B     555          BCS BUFEXX     ;SETAD > MAXBUF!
159Ø:          556 *
159Ø:AØ 7F     557 SETASC: LDY #$7F        ;Maske
1592:C9 A7     558          CMP #$A7       ;"'"
1594:FØ Ø6     559          BEQ SMASK
1596:C9 A2     56Ø          CMP #'"'       ;ASCII-MSB on?
1598:DØ 2D     561          BNE SETERR
159A:AØ FF     562          LDY #$FF
159C:8C D9 15  563 SMASK:   STY SETMASK
159F:E8        564          INX
15AØ:BD ØØ Ø2  565 SASCZ:   LDA CMDSTR,X
15A3:FØ 25     566          BEQ SETQT      ;CMDSTR zuende
15A5:C9 A1     567          CMP #STOP      ;ein "!"?
15A7:DØ Ø6     568          BNE SASC3      ;nein
15A9:DD Ø1 Ø2  569          CMP CMDSTR+1,X ;noch ein "!"?
15AC:DØ 1C     57Ø          BNE SETQT      ;nein, Ende
15AE:E8        571          INX            ;ja, => ein "!"
15AF:E8        572 SASC3:   INX            ;überspringen
15BØ:8E 9E ØC  573          STX CMDIDX
15B3:2D D9 15  574          AND SETMASK
15B6:2Ø F8 15  575          JSR SBYTE
15B9:9Ø E5     576          BCC SASCZ
15BB:          577 *
15BB:2Ø CA 15  578 BUFEXX:  JSR SETQT      ;Display der
15BE:AØ 5E     579          LDY #EXXBUF-ERRMSGS ;geänderten
15CØ:AE 9E ØC  58Ø          LDX CMDIDX     ;Bytes, danach:
15C3:CA        581          DEX
15C4:4C AC ØA  582          JMP DSPERR     ;"Buffer overflow"
15C7:          583 *
15C7:4C AA ØA  584 SETERR: JMP CMDERR
15CA:          585 *
15CA:AD C2 ØC  586 SETQT:   LDA SETAD      ;Adresse des letzten
15CD:8D A2 ØC  587          STA EADDR      ;veränderten Bytes+1
15DØ:AD C3 ØC  588          LDA SETAD+1    ;=> EADDR
15D3:8D A3 ØC  589          STA EADDR+1    ;danach Display der
15D6:4C 7D 12  59Ø          JMP REDUMP     ;geänderten Bytes
15D9:          591 *
15D9:          592 SETMASK: DS 1
15DA:          593 *
15DA:BD ØØ Ø2  595 GETHEX: LDA CMDSTR,X
15DD:FØ 14     596          BEQ GHEX3      ;CMDSTR zuende
15DF:C9 AØ     597          CMP #$AØ
15E1:DØ Ø3     598          BNE GHEXZ
15E3:E8        599          INX            ;<Space> wird
15E4:DØ F4     6ØØ          BNE GETHEX     ;übersprungen
15E6:8E 9E ØC  6Ø1 GHEXZ:   STX CMDIDX
15E9:2Ø 28 ØC  6Ø2          JSR EVALARG
15EC:BØ Ø6     6Ø3          BCS GOTHEX     ;ARG > $2ØØ
15EE:AD A7 ØC  6Ø4          LDA EVREG+1
15F1:FØ Ø1     6Ø5          BEQ GOTHEX
15F3:38        6Ø6 GHEX3:  SEC            ;Hex zu Ende
15F4:AD A6 ØC  6Ø7 GOTHEX: LDA EVREG
15F7:6Ø        6Ø8          RTS
15F8:          6Ø9 *
15F8:AC C3 ØC  61Ø SBYTE:   LDY SETAD+1
15FB:CC A3 ØC  611          CPY EADDR+1
15FE:AC C2 ØC  612          LDY SETAD      ;EADDR erreicht?
```

```
16Ø1:9Ø Ø5     613          BCC SBYT1
16Ø3:CC A2 ØC  614          CPY EADDR
16Ø6:BØ ØE     615          BCS SBYT2      ;ja, überspringen
16Ø8:99 ØØ 2Ø  616 SBYT1:   STA BUFFER,Y
16ØB:EE C2 ØC  617          INC SETAD
16ØE:DØ Ø6     618          BNE SBYT2
161Ø:EE C3 ØC  619          INC SETAD+1
1613:EE ØA 16  62Ø          INC SBYT1+2
1616:6Ø        621 SBYT2:   RTS            ;C set: SETAD=EADDR
1617:          622 *
1617:          625 ***************
1617:          626 *---COMPARE---*
1617:          627 ***************
1617:          628 *
1617:AE 9E ØC  629 COMPARE: LDX CMDIDX     ;nur für DSPERR
161A:2C BF ØC  63Ø          BIT HLSET      ;HOLD-Buffer set?
161D:1Ø 76     631          BPL HLDERR     ;nein, => HLDERR
161F:2Ø C5 ØB  632          JSR GETPRM     ;holt SADDR, EADDR
1622:2Ø FA 16  633          JSR LIMIT      ;EADDR auf SADDR+HLNUM
1625:8D B9 ØC  634          STA DSPTOP+1   ;+ Limit durch MAXBUF
1628:8C B8 ØC  635          STY DSPTOP     ;=> max. Display-Addr.
162B:2Ø 7Ø 12  636          JSR SADSTART   ;SADDR => DSPSTART
162E:8D B1 ØC  637          STA DSPBTM+1   ;SADDR ist untere
1631:8C BØ ØC  638          STY DSPBTM     ;Display-Grenze (SCRL)
1634:8D BC ØC  639          STA CPBASE+1   ;und Offset von HLBUF
1637:8C BB ØC  64Ø          STY CPBASE     ;zu BUFFER
163A:AØ Ø2     641          LDY #CMODE-AMODE ;Index zu CMODE
163C:A9 Ø8     642          LDA #$Ø8       ;8 Bytes per Zeile
163E:8C A8 ØC  643          STY NXTMODE    ;nächster Modus
1641:8D AD ØC  644          STA LBYTES
1644:A9 AB     645          LDA #'+'
1646:8D BA ØC  646          STA SUCCEED    ;SUCCEED auf '+'
1649:2C 9C ØC  647          BIT NODSP      ;Display OFF?
164C:3Ø Ø3     648          BMI CMPALL     ;ja, Gesamt-Vergleich
164E:4C 62 13  649          JMP SETMOD2    ;=> SETDUMP & DUMP
1651:          65Ø *
1651:AD B9 ØC  651 CMPALL: LDA DSPTOP+1 ;Vergleich des gesam-
1654:AC B8 ØC  652          LDY DSPTOP     ;ten HLBUF mit BUFFER
1657:8D B7 ØC  653          STA DSPEND+1   ;GNLINE bezieht sich
165A:8C B6 ØC  654          STY DSPEND     ;auf DSPEND!
165D:AD B1 ØC  655          LDA DSPBTM+1   ;Basis-Adresse
166Ø:AC BØ ØC  656          LDY DSPBTM
1663:8D B5 ØC  657          STA NXTADDR+1
1666:8C B4 ØC  658          STY NXTADDR
1669:2Ø 9E 16  659          JSR SCPIDX     ;setzt CPINDEX
166C:AD AD ØC  66Ø          LDA LBYTES     ;Anzahl Bytes/Zeile
166F:38        661          SEC
167Ø:E9 Ø1     662          SBC #1
1672:8D FC 14  663          STA CRMASK
1675:          664 *
1675:2Ø B5 14  665 CPA1:    JSR GNLINE     ;holt eine "Zeile" aus
1678:A2 ØØ     666          LDX #ØØ        ;BUFFER & setzt LTOP
167A:2Ø 22 15  667 CPA2:    JSR GCBYTE     ;ein Byte aus HLBUF
167D:DD Ø2 15  668          CMP LBUF,X
168Ø:DØ ØD     669          BNE NOTSAME    ;=> SUCCEED auf "-"
1682:E8        67Ø          INX
1683:EC Ø1 15  671          CPX LTOP       ;"Zeilenende"?
1686:9Ø F2     672          BCC CPA2
1688:2C ØØ 15  673          BIT DDONE      ;gesetzt von GNLINE
168B:1Ø E8     674          BPL CPA1
168D:3Ø Ø5     675          BMI CPQUIT     ;HLBUF=BUFFER (=>"+")
168F:A9 AD     676 NOTSAME: LDA #'-'
1691:8D BA ØC  677          STA SUCCEED
1694:6Ø        678 CPQUIT: RTS
1695:          679 *
1695:AØ 72     68Ø HLDERR: LDY #NOBUF-ERRMSGS
1697:AE 9E ØC  681          LDX CMDIDX
169A:CA        682          DEX
169B:4C AC ØA  683          JMP DSPERR     ;"No Hold Buffer set"
169E:          684 *
169E:AD B4 ØC  685 SCPIDX: LDA NXTADDR     ;Berechnung von CPIDX
16A1:38        686          SEC            ;via Offset zu BUFFER
16A2:ED BB ØC  687          SBC CPBASE
16A5:8D BD ØC  688          STA CPIDX      ;für SCROLL und
16A8:AD B5 ØC  689          LDA NXTADDR+1  ;REDUMP notwendig!
16AB:ED BC ØC  69Ø          SBC CPBASE+1
16AE:8D BE ØC  691          STA CPIDX+1
16B1:6Ø        692          RTS
16B2:          693 *
16B2:          696 *************
16B2:          697 *---HOLD---*
16B2:          698 *************
16B2:          699 *
16B2:2Ø C5 ØB  7ØØ HOLD:    JSR GETPRM
16B5:AD A1 ØC  7Ø1          LDA SADDR+1    ;SADDR => NXTADDR
16B8:AC AØ ØC  7Ø2          LDY SADDR
16BB:8D B5 ØC  7Ø3          STA NXTADDR+1  ;GNBYTE läuft von
```

```
16BE:8C B4 ØC  7Ø4          STY NXTADDR    ;NXTADDR..DSPEND
16C1:2Ø 4E 12  7Ø5          JSR SAVEND     ;DSPEND retten
16C4:8D B7 ØC  7Ø6          STA DSPEND+1   ;returnt EADDR, im
16C7:8C B6 ØC  7Ø7          STY DSPEND     ;Gegensatz zu REDUMP
16CA:A9 22     7Ø8          LDA #<HLBUF    1 kein Limit auf DSPTOP
16CC:8D D9 16  7Ø9          STA TOHL+2     ;Zieladresse
16CF:A2 ØØ     71Ø          LDX #ØØ        ;Zähler
16D1:8E C1 ØC  711          STX HLNUM+1
16D4:2Ø CC 14  712 FROMBUF: JSR GNBYTE
16D7:9D ØØ 22  713 TOHL:    STA HLBUF,X    ;modifiziert...
16DA:E8        714          INX
16DB:DØ Ø6     715          BNE HLD2
16DD:EE D9 16  716          INC TOHL+2     ;Zieladresse
16EØ:EE C1 ØC  717          INC HLNUM+1    ;Zähler
16E3:AD ØØ 15  718 HLD2:    LDA DDONE      ;EADDR erreicht?
16E6:1Ø EC     719          BPL FROMBUF
16E8:8E CØ ØC  72Ø          STX HLNUM      ;Zähler-Lowbyte
16EB:8D BF ØC  721          STA HLSET      ;Flag: HOLD-Buffer set
16EE:2Ø 61 12  722          JSR RESEND     ;Restore von DSPEND
16F1:2C AB ØC  723          BIT CMODE      ;nur bei CMODE muß ein
16F4:1Ø Ø3     724          BPL HLDONE     ;REDUMP stattfinden
16F6:2Ø 7D 12  725          JSR REDUMP
16F9:6Ø        726 HLDONE: RTS
16FA:          727 *
16FA:AD AØ ØC  729 LIMIT:   LDA SADDR      ;die Endadresse für
16FD:18        73Ø          CLC            ;COMPARE und MOVE ist
16FE:6D CØ ØC  731          ADC HLNUM      ;SADDR+HLNUM. Wenn
17Ø1:A8        732          TAY            ;MAXBUF überschritten
17Ø2:AD A1 ØC  733          LDA SADDR+1    ;wird, dann Limit
17Ø5:6D C1 ØC  734          ADC HLNUM+1    ;auf MAXBUF.
17Ø8:C9 Ø2     735          CMP #MAXBUF
17ØA:9Ø ØA     736          BCC TOPOK      ;SADDR+HLNUM<MAXBUF
17ØC:DØ Ø4     737          BNE LIMTOP
17ØE:CØ ØØ     738          CPY #ØØ
171Ø:FØ Ø4     739          BEQ TOPOK
1712:A9 Ø2     74Ø LIMTOP: LDA #MAXBUF    ;Limit auf MAXBUF
1714:AØ ØØ     741          LDY #ØØ
1716:CD A3 ØC  742 TOPOK:   CMP EADDR+1    ;geht die EADDR über
1719:9Ø Ø7     743          BCC LIMEAD     ;SADDR+HLNUM hinaus?
171B:DØ ØB     744          BNE LMDONE
171D:CC A2 ØC  745          CPY EADDR
172Ø:BØ Ø6     746          BCS LMDONE
1722:8D A3 ØC  747 LIMEAD: STA EADDR+1     ;ja, Limit von EADDR
1725:8C A2 ØC  748          STY EADDR      ;auf SADDR+HLNUM
1728:6Ø        749 LMDONE: RTS            ;mit DSPTOP (f. CMP)
1729:          75Ø *
1729:          752 ************
1729:          753 *---MOVE---*
1729:          754 ************
1729:          755 *
1729:2C BF ØC  756 MOVE:    BIT HLSET
172C:3Ø Ø3     757          BMI MVØ
172E:4C 95 16  758          JMP HLDERR     ;"No Hold Buffer set"
1731:2Ø C5 ØB  759 MVØ:     JSR GETPRM
1734:2Ø FA 16  76Ø          JSR LIMIT      ;Begrenzung auf MAXBUF
1737:AD A1 ØC  761          LDA SADDR+1    ;oder HLBUF-Ende
173A:8D 74 17  762          STA MVADDR     ;Hi-Adresse für MOVE
173D:18        763          CLC
173E:69 2Ø     764          ADC #<BUFFER
174Ø:8D 52 17  765          STA TOBUF+2    ;Ziel-Adresse
1743:A9 22     766          LDA #<HLBUF
1745:8D 4F 17  767          STA FROMHL+2   ;Quelle
1748:A2 ØØ     768          LDX #Ø
174A:AC AØ ØC  769          LDY SADDR
174D:BD ØØ 22  77Ø FROMHL: LDA HLBUF,X
175Ø:99 ØØ 2Ø  771 TOBUF:   STA BUFFER,Y
1753:C8        772          INY
1754:DØ Ø6     773          BNE MV1
1756:EE 52 17  774          INC TOBUF+2    ;Ziel + $1ØØ
1759:EE 74 17  775          INC MVADDR     ;MOVE-Seitenzähler
175C:CC A2 ØC  776 MV1:     CPY EADDR
175F:DØ Ø8     777          BNE MV2
1761:AD 74 17  778          LDA MVADDR
1764:CD A3 ØC  779          CMP EADDR+1
1767:BØ Ø8     78Ø          BCS MVDONE     ;EADDR ereicht
1769:E8        781 MV2:     INX
176A:DØ E1     782          BNE FROMHL
176C:EE 4F 17  783          INC FROMHL+2   ;Quelle + $1ØØ
176F:DØ DC     784          BNE FROMHL
1771:4C 7D 12  785 MVDONE: JMP REDUMP      ;Neu-Anzeige
1774:          786 *
1774:          787 MVADDR: DS 1            ;Seitenzähler
1775:          788 *
1775:          789          CHN ED.FUNCS2.TXT
```

ED.FUNCS2.TXT2

```
1775:4C AA ØA    1 GETVAR: JMP CMDERR
1778:            2 *
1778:            3          CHN ED.CMDS.TXT
```

ED.CMDS.TXT2

```
1778:            1          PAGE
1778:            2 ****************
1778:            3 *---COMMANDS---*
1778:            4 ****************
1778:            5 *
1778:            6 CMDS:    EQU *
1778:CØ          7          ASC '§'        ;Lädt den ersten Block
1779:28 ØF       8          DW  ATFILE     ;eines Files
177B:D8          9          ASC 'X'        ;MONITOR-Aufruf
177C:F6 1Ø      1Ø          DW  XMON
177E:D1         11          ASC 'Q'        ;Programm-Ende
177F:4F 11      12          DW  QUIT
1781:D2         13          ASC 'R'        ;Read Block/Prefix
1782:           14 *
1782:9B ØD      15          DW  READ
1784:D7         16          ASC 'W'        ;Write Block/Prefix
1785:A3 ØD      17          DW  WRITE
1787:           18 *
1787:D3         19          ASC 'S'
1788:3A 15      2Ø          DW  SET        ;Bytes verändern
178A:           21 *
178A:C1         22          ASC 'A'
178B:36 13      23          DW  DUMPA      ;ASCII-Dump
178D:C4         24          ASC 'D'
178E:3C 13      25          DW  DUMPH      ;Hex/ASCII-Dump
179Ø:D6         26          ASC 'V'
1791:42 13      27          DW  DUMPV      ;VBM-Dump
1793:           28 *
1793:BC         29          ASC '<'
1794:58 11      3Ø          DW  HALFUP     ;halbe Seite aufwärts
1796:AC         31          ASC ','
1797:63 11      32          DW  LINEUP     ;Zeile aufwärts
1799:BE         33          ASC '>'
179A:94 11      34          DW  HALFDN     ;halbe Seite abwärts
179C:AE         35          ASC '.'
179D:9F 11      36          DW  LINEDN     ;Zeile abwärts
179F:           37 *
179F:C8         38          ASC 'H'
17AØ:B2 16      39          DW  HOLD       ;HLBUF setzen
17A2:CD         4Ø          ASC 'M'
17A3:29 17      41          DW  MOVE       ;aus HLBUF kopieren
17A5:C3         42          ASC 'C'        ;Vergleich
17A6:17 16      43          DW  COMPARE    ;BUFFER mit HLBUF
17A8:           44 *
ØØ3Ø:           46 LCMD:   EQU *-CMDS
17A8:           47 *
17A8:           5Ø ******************************
17A8:           51 *
Ø2ØØ:           52 BUFLEN: EQU MAXBUF*$1ØØ
17A8:           53 *
2ØØØ:           54 BUFFER: EQU $2ØØØ
22ØØ:           55 HLBUF: EQU BUFFER+BUFLEN
24ØØ:           56 FNBUF: EQU HLBUF+BUFLEN
25ØØ:           57 FNMASK: EQU FNBUF+$1ØØ
26ØØ:           58 EDITEND: EQU FNMASK+$1ØØ
17A8:           59 *
17A8:           6Ø         LST OFF
```

ProDOS-Runtime-Library

Eine Muster-Bibliothek für Assembler-, Applesoft- und Pascal-Programme

von Ulrich Stiehl

1. Problemstellung

Der sog. ProDOS-Kern, der übrigens auf der ursprünglichen Entwickler-Diskette noch unter dem Namen „KERNEL" (heute „PRODOS") gespeichert war, belegt die Bank 1 der Language-Card ($D000-$FFFF; Softswitch: $C08B). Hinzu kommen noch die Global-Page ($BF00-$BFFF) sowie die normalerweise entbehrliche Reboot-Routine in der Bank 2 der Language-Card ($D100-$D3FF; Softswitch: $C083). Damit wäre der Speicherbereich $0800-$BEFF theoretisch frei für eigene Programme und Daten. Es fehlt jedoch jetzt noch ein Verbindungsprogramm, das Befehle an die MLI-Schittstelle weitergibt (MLI = Machine Language Interface). Das bekannteste Verbindungsprogramm ist das BASIC.SYSTEM, das jedoch erstens nur für nicht-compilierte Applesoft-Programme gedacht ist und zweitens mit nur einem I/O-Puffer den Bereich $9600-$BEFF und damit genausoviel Speicher wie das *gesamte* DOS 3.3 mit drei I/O-Puffern belegt.

Nachfolgend wollen wir eine kleine Muster-Bibliothek für ProDOS vorstellen, die als Verbindungsprogramm für beliebige Anwenderprogramme (Assembler, Applesoft, Kyan-Pascal usw.) eingesetzt werden kann. Diese Bibliothek dient zugleich als Beispiel für eine modulare und damit flexibel einsetzbare Programmsammlung:
– Die Routinen verändern keine Prozessor-Register und keine Zero-Page-Speicherstellen und sind damit für das aufrufende Hauptprogramm völlig transparent.
– Jeder der implementierten Befehle (Online, Prefix usw.) hat nur einen einzigen Eingang und Ausgang und kann beliebig erweitert, gekürzt oder ganz entfernt werden.
– Für das Einfügen von zusätzlichen Befehlen, Fehlermeldungen usw. sind bereits Vorkehrungen getroffen worden.

2. Befehlsübersicht

Insgesamt umfaßt die Runtime-Library folgende Befehle:
0 = Reboot: Aufruf der Reboot-Routine (entspricht dem BYE-Befehl unter dem BASIC.SYSTEM 1.1) – parameterlose Prozedur.
1 = Online: Zeigt die Namen aller angeschlossenen Volumes mit den entsprechenden Slot/Drive-Parametern sowie zusätzlich das momentane Präfix (= Default-Prefix = Ersatz-Präfix) an – parameterlose Prozedur.
2 = Prefix: Setzt entweder ein neues Präfix (Set Prefix) oder zeigt das momentane Präfix an (Get Prefix) – Parameter: Directory-Name (= NAME0).
3 = Cat: Zeigt das spezifizierte oder das Default-Directory an – Parameter: Directory-Name (= NAME0 oder NAME1).
4 = Create: Erzeugt ein Subdirectory – Parameter: Directory-Name (= NAME1).
5 = Delete: Löscht eine Datei oder ein (leeres) Directory – Parameter: Name (= NAME1).
6 = Lock: Setzt den Schreibschutz („Sternchen") einer Datei oder eines Directory – Parameter: Name (= NAME1).
7 = Unlock: Entfernt den Schreibschutz einer Datei oder eines Directory – Parameter: Name (= NAME1).
8 = Rename: Ändert den Namen einer Datei oder eines Directory – Parameter: Altname (= NAME1) und Neuname (= NAME2).
9 = Bsave: Speichert einen durch Startadresse und Länge definierten Speicherbereich als BIN-Datei – Parameter: Dateiname (= NAME1), Bsave-Anfangsadresse (= BANF) und Bsave-Länge (= BLEN).
10 = Bload: Lädt eine BIN-Datei in den Speicher ab einer vorgegebenen Startadresse unter Beachtung der maximal zulässigen Länge (= Endadresse minus Startadresse) – Parameter: Dateiname (NAME1), Bload-Anfangsadresse (= BANF) und maximale Bload-Länge (= BLEN).

Unsere auf die grundlegenden Befehle beschränkte ProDOS-Runtime kann leicht durch speziellere Befehle erweitert werden. Für eine Applesoft-Runtime würden sich beispielsweise die zusätzlichen Befehle Run, Load und Save anbieten, die sich kaum von den obigen Bload- und Bsave-Befehlen unterscheiden würden. Ferner könnte man eine Ein/Ausgaberoutine für das Datum und die Uhrzeit (= Date) einbauen. Außerdem wäre ein CP/M-Type-Befehl (= Hex/ASCII-Dump-Befehl) als Abwandlung des Bload-Befehls denkbar. Schließlich könnte man

Routinen zur Bearbeitung von Textfiles im allgemeinen und Random-Access-Files im besonderen entwickeln.

3. PRODOS.LIB

Die nachfolgende Beschreibung bezieht sich auf das Listing PRODOS.LIB, wobei „Z" für „Zeile" steht.

3.1. Eingabe-Parameter

BPUF (siehe Z. 29ff.) ist der I/O-Block-Puffer, der 2 Blöcke zu je 512 Bytes umfassen muß, weil ProDOS neben dem eigentlichen Datenblock zusätzlich den jeweiligen Index-Block verwaltet. BPUF muß ferner im Gegensatz zu DPUF auf einer Seitengrenze beginnen.
DPUF ist der allgemeine Datenpuffer (hier 512 Bytes), der von der PRODOS.LIB für die Befehle Online und Cat benötigt wird. Für Online allein würden 256 Bytes genügen, doch für Cat ist die Größe von 512 Bytes unabdingbar, weil dieser Befehl sonst zu langsam wäre. Das gleiche würde für den Type-Befehl gelten. (Unsere Cat-Routine setzt einen 512-Byte-Datenpuffer auf Seitengrenze voraus. Im Falle eines kleineren Datenpuffers muß der Cat-Algorithmus umgeschrieben werden.)
BPUF und DPUF müssen nicht gepokt werden, sondern sind bereits im Assembler-Quelltext verankert.

NAME0, NAME1 und NAME2 sind die Präfix-, Datei- und Rename-Namen, die jeweils 1 Länge-Byte sowie 64 ASCII-Bytes belegen und in dieser Form an die ProDOS-Runtime übergeben werden müssen.
Als Ausnahme sind bei NAME0 für Get Prefix nicht 1 + 64 = 65 Bytes, sondern 1 + (64 + 2) = 67 Bytes reserviert worden, weil infolge einer Eigentümlichkeit des MLI im Extremfall ein bis zu 66 Zeichen langes ASCII-Präfix über Get Prefix zurückgeliefert werden kann, das aber dann von unserer Set-Prefix-Routine sicherheitshalber sofort auf Null gesetzt wird.
Sind Sie experimentierfreudig? Dann geben Sie folgendes Applesoft-Programm ein:

```
1 PRINT CHR$(4);"PREFIX/RAM":
  REM 3stelliger Name erforderlich!
2 X$ = "XXXX": 4stelliger Name!
3 FOR X = 1 TO 12
4 PRINT CHR$(4);"CREATE"X$
5 PRINT CHR$(4);"PREFIX"X$
6 NEXT
7 PRINT CHR$(4);"PREFIX"
8 INPUT X$: PRINT X$
9 PRINT CHR$(4);"CAT": REM Hier!
```

In Zeile 9 bricht das Programm mit einem Syntax-Error ab, weil das Präfix jetzt mehr als 64 Zeichen umfaßt. Es wird zwar in Zeile 7-8 gerade noch korrekt über Get Prefix angezeigt, doch führt nunmehr jeder andere Befehl (hier Cat) zu einer Fehlermeldung. Das BASIC.SYSTEM wird damit praktisch lahmgelegt.

BANF und BLEN stehen für Bload/Bsave-Anfangsadresse und Bload/Bsave-Länge, die im Low-High-Format an die ProDOS-Runtime übergeben werden müssen.
BEFEHL ist die Nummer für den jeweiligen Befehl (0-10, s.o.). Wenn man in Z. 144 „CMP #10+1" durch z.B. „CMP #1X+1" ersetzt, so kann man die Sprungtabelle (Z. 158-168) nach Z. 168 durch X Befehle erweitern (X steht hier für eine Zahl im Bereich 1 bis 9). Umgekehrt kann die Befehlstabelle bei Bedarf entsprechend gekürzt werden.

ENTRY (Z. 133): Wenn die Befehlsnummer sowie die für diesen Befehl erforderlichen Eingabe-Parameter gepokt worden sind, erfolgt der Aufruf durch JSR ENTRY (Assembler) oder CALL ENTRY (Applesoft) oder in Kyan-Pascal durch eine entsprechende Prozedur (s.u.). Es sei darauf hingewiesen, daß die Eingabe-Parameter von der ProDOS-Runtime nicht verändert werden (Ausnahme: NAME0 mit mehr als 64 ASCII-Zeichen bei Get Prefix), so daß konstante Eingabe-Parameter nur einmal gepokt werden müssen.

3.2. Ausgabe-Parameter

EXIT (Z. 173-236): Die ProDOS-Runtime wird stets über EXIT verlassen, und zwar letztlich in Z. 236 („RTS"). Zuvor werden jedoch noch mögliche Fehler ausgewertet.

ERRMLI (siehe Z. 29 ff.) enthält die MLI-Fehlernummer (0 = kein Fehler), die als ASCII-Meldung ausgegeben wird, wenn sie in der Fehlertabelle (Z. 243-278) enthalten ist. Diese Tabelle kann je nach Speicherkapazität beliebig erweitert und gekürzt werden. Nehmen wir an, wir wollten die MLI-Fehlernummer $4C = „End of File bei Read überschritten" zusätzlich einbauen. Dann fügen wir vor Z. 278 (= Endmarker) folgende Zeilen ein:

```
HEX 4C            ;Nummernzeile
HEX ØØ            ;Ø-Marke
ASC "End of..."   ;Text
HEX ØØ            ;Ø-Marke
```

In der Nummernzeile (vgl. Z. 243, 248, 258, 263 und 268) können auch mehrere Fehlernummern zu gleichartigen ASCII-Meldungen zusammengefaßt werden. Der Suchalgorithmus (Z. 179-217) ist so angelegt, daß die Fehlertabelle beliebig groß sein kann. Würde man jedoch alle MLI-Fehler mit ausführlichen Meldungen berücksichtigen, dann würde die Fehlertabelle mehr Speicherraum als die restliche ProDOS-Runtime einnehmen. Eine Auswahl-Fehlertabelle ist daher wohl am zweckmäßigsten. Wenn Sie jedoch noch mehr Platz sparen müssen, so entfernen Sie Z. 179-182 (183 bleibt!), Z. 184-217 und Z. 243-278. Dann wird nur eine hexadezimale Fehlernummer in der Form „MLI-Fehler Nr. XX" ausgegeben.
ERRALL ist eine allgemeine Fehlernummer (siehe Z. 114-119: $80-$85; 0 = kein Fehler), die vom aufrufenden Hauptprogramm (oder genauer gesagt von Ihnen selbst!) verarbeitet werden muß, denn ERRLL ist nur dann <> 0, wenn Sie selbst einen Fehler gemacht haben (und nicht der Anwender Ihres Programms).

FLEN enthält nach Ausführung des Bload-Befehls die Anzahl der effektiv eingelesenen Bytes. FLEN spielt nur dann eine Rolle, wenn FLEN > BLEN ist (ERRALL $85).

3.3. Standard-MLI-Aufrufe

Z. 283-480 umfaßt *alle* MLI-Aufrufe mit Ausnahme der Block- und Interrupt-Befehle, wobei die Muster-Bibliothek die MLI-Aufrufe Flush (Z. 426-431), Set Mark (Z. 433-439), Get Mark (Z. 441-447), Set EOF (449-455), Set Buffer (Z. 465-471) und Get Buffer (Z. 473-479) selbst nicht benutzt. Diese letzteren Befehle würden z.B. benötigt, wenn man Random-Access-Dateien bearbeiten wollte.
Bei einigen MLI-Parametern, im Listing gekennzeichnet durch „0 (1)", können von den Runtime-Routinen (Z. 497-1139) auch andere Werte gepokt werden. Dies gilt insbesondere für den Close-Befehl (Z. 424), bei dem CLREF = 0 *alle* offenen Dateien schließt, was im Falle von sog. Open-Files unerwünscht wäre.

3.4. Hex- und ASCII-Ausgabe

Die Ausgaberoutinen für Online, Prefix (Get Prefix) und Cat sowie die Fehlermeldungen (Z. 486-494) wurden derart umgelenkt, daß die zwei einzigen Monitorroutinen COUT (ASCII-Ausgabe) und PRBYTE (Hexbyte-Ausgabe) nur in den Zeilen 491 und 494 auftauchen, denn nur durch diese zwei Monitoraufrufe können von der ProDOS-Runtime indirekt Zero-Page-Adressen verändert werden.
Damit ergibt sich zugleich ein weiterer Vorteil: Wenn die Texte nicht auf den Bildschirm ausgegeben, sondern z.B. an eine Applesoft- oder Kyan-Pascal-String-Variable zurückgeliefert werden sollen, so kann man in Z. 486-494 eine entsprechende Übergaberoutine einbauen.

3.5. Runtime-Befehle

Die Runtime-Befehle – von Reboot bis Bload (Z. 497-1139) – lassen verschiedene Erweiterungen zu. Dies gilt namentlich für den Cat-Befehl, dessen Ausgabeformat (Z. 677-679) in Anlehnung an DOS 3.3 auf das absolute Minimum reduziert worden ist. Wenn Sie Änderungen oder Erweiterungen vornehmen wollen, so sollten Sie jedoch vorher den Quelltext des entsprechenden Befehls genau studieren. Wenn Sie umgekehrt einen Befehl streichen wollen, so geht dies auch ohne Assemblerkenntnisse. Nehmen wir an, die Befehle Bload und Bsave seien entbehrlich. Dann löschen wir Z. 167-168 sowie Z. 955-1139 und ersetzen Z. 144 durch „CMP #8+1". Fertig!

Kleine Erweiterungen
Wenn man in Z. 364 (= ONUNIT) z.B. $60 (bzw. $E0) pokt, dann wird der *Online-Befehl* nur in bezug auf S6, D1 (bzw. S6, D2) ausgeführt.
Beim *Cat-Befehl* kann man die Anzahl der von einer Datei belegten Blocks als Dezimal- statt Hexadezimalzahl ausgeben, wenn der Applesoft-Interpreter verfügbar ist, denn dann braucht man nur Z. 786 („JSR HEXAX") durch „JSR $ED24" (LINPRT-Routine) zu ersetzen.
Den *Create-Befehl* kann man durch Poken des Datei- und Speichertyps in Z. 838-839 auch zur Anlage von Nicht-Directory-Files verwenden.
Um die *Bload/Bsave-Befehle* auch bei Nicht-BIN-Files, z.B. TXT-Files, verwenden zu können, braucht man nur in Z. 967 bzw. Z. 1056 den Dateityp zu poken.

4. PRODOS.LIB.DEMO

Dieses Demo veranschaulicht die Anwendung der Runtime-Library unter Applesoft. In Z. 10 wird das BASIC.SYSTEM mit
POKE 242, 0: PR#0: IN#0
abgehängt, so daß Sie nach Beendigung des Demos Ctrl-Reset drücken müssen, um die Ein/Ausgabevektoren des BASIC. SYSTEMs wieder anzuschließen.
Beachten Sie, daß die ProDOS-Runtime ein Ersatz-Präfix (Default-Prefix) zwingend voraussetzt. Wenn man eine ProDOS-Diskette bootet, die nur die Dateien PRODOS und BASIC.SYSTEM enthält, so wird *kein* Ersatz-Präfix erzeugt. Daß die BASIC. SYSTEM-Befehle CAT usw. trotzdem funktionieren, liegt daran, daß die Ersatz-Slot/Drive-Parameter auf z.B. S6,D1 gesetzt wurden. Unsere ProDOS-Runtime unterstützt jedoch keine Slots und Drives. Folglich müssen Sie notfalls vor
RUN PRODOS.LIB.DEMO
mit z.B.
PREFIX, S6,D1
ein Ersatz-Präfix festlegen. Geschickter verfahren Sie, wenn jede ProDOS-Diskette ein STARTUP-Programm enthält, in dessen erster Programmzeile mit z.B.
1 PRINT CHR$(4);"PREFIX": INPUT P$: PRINT CHR$(4); "PREFIX";P$
das Ersatz-Präfix festgelegt wird.

5. PRODOS.LIB.P

Dieses Demo zeigt, wie das Rahmenprogramm zu unserer ProDOS-Runtime unter Kyan-Pascal aussehen könnte. Sie können jedoch den Quelltext T.PRODOS.LIB nicht ohne weiteres zu einem Kyan-Pascal-Include-File deklarieren, weil die Syntax des Kyan-Assemblers von dem des Big-Mac-Assemblers erheblich abweicht. Insbesondere müssen alle Label-Namen auf 6 Stellen gekürzt werden. Ferner müssen die HEX- und DA-Zeilen in DB- und DW-Zeilen (unter Berücksichtigung des $-Zeichens!) umgeändert, die Low/High-Byte-Kürzel („<", „>") umgedreht und die String-Begrenzer von " in ' abgewandelt werden. Schließlich muß die ORG-Zeile entfernt und zugleich dafür Sorge getragen werden, daß BPUF unter allen Umständen auf einer Seitengrenze beginnt. Auf der Kyan-Club-Diskette #A, die allerdings wegen der Systemdateien nur für Club-Mitglieder erhältlich ist, befinden sich verschiedene und um einen Kopierbefehl erweiterte Kyan-Varianten der PRODOS.LIB unter den Namen FID.P, FID1.P und FID2.P.
Abschließend möchte ich darauf hinweisen, daß Arne Schäpers im Anschluß an seine BASIC.SYSTEM-Analyse (= Pendant zur ProDOS-Analyse), die in Kürze fertiggestellt sein wird, ein kompaktes und für den TASC-Compiler geeignetes neues BASIC.SYSTEM schreiben wird, das je nach Umfang entweder im Peeker oder im Hüthig-Software-Service erscheinen wird.

Kurzhinweise

1. Zweck: ProDOS-Runtime-Bibliothek für Assembler-, Applesoft- und Kyan-Programme.
2. Konfiguration:
Apple II+/e/c; ProDOS (alle Versionen bis 1.1.1)
3. Test:
RUN PRODOS.LIB.DEMO
4. Sammeldisk:
T.PRODOS.LIB (Big-Mac-Quelltext)
PRODOS.LIB (Objektcode)
PRODOS.LIB.DEMO
5. Sonstiges:
Die Dateien müssen zunächst von der DOS-3.3-Sammeldisk mit CONVERT oder DOSTOPRO auf Ihre ProDOS-Arbeitsdiskette konvertiert werden.

Z 8 0 + Card
Die leistungsfähige und vielseitige Z80 Karte für Ihren Apple II

Hardware:
+ ALS CP/M-Card- **und** (!) SoftCard-kompatibel.
+ 8MHz Taktfrequenz (ohne Waitzyklen) – dadurch werden Programme bis zu 4x schneller als bisher (2MHz Taktfrequenz im SoftCard-Modus).
+ Interrupts sind für die beiden Betriebsmodi der Z80+ Card getrennt zu- und abschaltbar

Software:
+ CP/M 2.20/2.23 Pseudodisktreiber, der die eigenen 64K RAM der Z80+ Card im SoftCard-Modus nutzt.
+ CP/M-Patch generiert aus Ihrem SoftCard CP/M 2.20 ein CP/M 2.2 für den schnellen Taktmodus der Z80+ Card mit überragenden Eigenschaften:
Bis zu 4x schnellerer Disk-Zugriff (Read), integrierte Pseudodisk mit 32/48/110K bei einem Apple mit 48/64/128K RAM, Unterstützung von Disk-Laufwerken bis 640K an den verschiedensten Controllern, Unterstützung der gängigsten RAM-Karten als Pseudodisk, schnelle Bildschirmausgabe, bedienerfreundlich durch neue und erweiterte Befehle in der Kommandoebene, 58K TPA.
+ **Der Preis: 696,-- DM** (= 610,53 DM ohne Mwst.)

Uwe Zimmermann – Microcomputerentwicklungen
Schwalbenstraße 30 – 6090 Russelsheim
Telefon: 06142 / 56 34 56

erphi rq

640KB Controller und Floppy für Apple II+/e und kompatible.

Generalvertretung für die Schweiz und **FL**

(Händleranfragen erwünscht)

beltronic

Im Chapf CH-8455 Rüdlingen
Tel: 01-8 67 31 41 · Tlx 8 25 981

Weiter führen wir:
IBM kompatible PC ab 2400,–
Apple IIe kompatible ab 1200,–

Ältere Peeker-Hefte

können für DM 6,50 pro Heft zuzüglich Versandspesen angefordert werden. Vergriffene Hefte sind als Photokopien für DM 10,– pro Heft erhältlich. Mindestbestellmenge 2 Hefte.

Dr. A. Hüthig Verlag · Heidelberg

PRODOS.LIB

```
BSAVE PRODOS.LIB, A$4000, L$0D59 (mit Pufferseiten)
BSAVE PRODOS.LIB, A$4600, L$0759 (ohne Pufferseiten)

                 1              ORG  $4000
                 2    *
                 3    * PRODOS.LIB U.Stiehl/10.5.86
                 4    * ==========
                 5    *
                 6    * Aufrufen mit
                 7    * LDA #Befehlsnummer
                 8    * STA BEFEHL
                 9    * JSR ENTRY
                10    *
                11    * Ferner zuvor befehlsspezifische
                12    * Übergabeparameter setzen (s.u.)
                13    *
                14    * Die Routine kehrt zurück mit
                15    * Fehler-Nr. in ERRMLI + ERRALL.
                16    * Wenn beide 00, dann alles okay.
                17    *
                18    * Die Register A, X, Y sowie
                19    * Zero-Page IND und IND+1 werden
                20    * automatisch gerettet.
                21    *-------------------------------
                22    IND      EQU  $0000       ;-$0001
                23    MLI      EQU  $BF00
                24    MLIDAT   EQU  $BF90
                25    MLIUHR   EQU  $BF92
                26    COUT     EQU  $FDED
                27    PRBYTE   EQU  $FDDA
                28    *-------------------------------
                29    BPUF     DS   $0400       ;Seiten-
                30    DPUF     DS   $0200       ;grenze!
4600: 00        31    NAME0    HEX  00          ;Länge
                32    ASCII0   DS   66          ;Ascii
4643: 00        33    NAME1    HEX  00          ;Länge
                34    ASCII1   DS   64          ;Ascii
4684: 00        35    NAME2    HEX  00          ;Länge
                36    ASCII2   DS   64          ;Ascii
46C5: 00 00     37    BANF     HEX  0000        ;LLHH
46C7: 00 00     38    BLEN     HEX  0000        ;LLHH
46C9: 00 00     39    FLEN     HEX  0000        ;Bload
46CB: 00        40    BEFEHL   HEX  00          ;0-10
46CC: 00        41    ERRMLI   HEX  00          ;0=okay
46CD: 00        42    ERRALL   HEX  00          ;0=okay
                43    *-------------------------------
                44    * BPUF   = Block-Puffer     (1024)
                45    * DPUF   = Daten-Puffer      (512)
                46    * NAME0  = Länge + Präfix     (67)
                47    * NAME1  = Länge + Name 1     (65)
                48    * NAME2  = Länge + Name 2     (65)
                49    * BANF   = LLHH Bsave-Anfang  (2)
                50    * BLEN   = LLHH Bsave-Länge   (2)
                51    * FLEN   = LLHH File-Länge    (2)
                52    * BEFEHL = 1-10               (1)
                53    * ERRMLI = Error MLI          (1)
                54    * ERRALL = Error allgemein    (1)
                55    *
                56    * BPUF = Block-Puffer=I/O-Puffer
                57    * wird von den OPEN-abhängigen
                58    * Befehlen benutzt, also von
                59    * Cat, Bsave und Bload.
                60    *
                61    * DPUF = Daten-Puffer wird von
                62    * Online und Cat benutzt.
                63    *
                64    * Die Block- und Daten-Puffer
                65    * müssen auf einer Seitengrenze
                66    * beginnen (hier z.B. $4000).
                67    *
                68    * NAME0 ist das Default-Präfix,
                69    * das mit "/" beginnen muß und
                70    * mit NAME1 oder NAME2 gekoppelt
                71    * wird, falls letztere nicht mit
                72    * "/" anfangen. NAME2 kommt
                73    * nur bei Rename vor. Ansonsten
                74    * wird immer NAME1 verwendet bei
                75    * Cat, Create, Delete, Lock,
                76    * Unlock, Bsave und Bload.
                77    *
                78    * WARNUNG: Sie müssen selbst
                79    * dafür sorgen, daß NAME0, NAME1
                80    * und NAME2 <= 64 sind und nicht
                81    * mehr als 64 Ascii-Zeichen ein-
                82    * getragen werden. Andernfalls
                83    * dreht das Programm durch!
                84    * Es wird nur geprüft, ob das
                85    * jeweilige Länge-Byte = 0 ist.
                86    *
                87    * BANF = Bsave-Anfang und
                88    * BLEN = Bsave-Länge enthalten
                89    * Anfangsadresse und Länge des
                90    * Speicherbereichs, der auf
                91    * Diskette gespeichert oder von
                92    * dort geladen werden soll.
                93    *
                94    * FLEN = File-Länge, die nach-
                95    * träglich vom BLOAD-Befehl
                96    * gepokt wird.
                97    * Wenn FLEN > BLEN, so wird
                98    * nur BLEN eingelesen und in FLEN
                99    * eingetragen (ERRALL = $85)
               100    *
               101    * Bei BSAVE bedeutet BANF+BLEN
               102    * das exakte Speicherende.
               103    * Bei BLOAD bedeutet BANF+BLEN
               104    * das maximale Speicherende.
               105    *
               106    * WARNUNG: Bei Bsave/Bload wird
               107    * nur geprüft, ob BLEN > 0 ist.
               108    * Bitmap-Konflikte müssen Sie
               109    * selbst abfangen!
               110    *
               111    * ERRMLI: MLI-Fehler: $00-$5A
               112    * ERRALL: Allgemeine Fehler:
               113    *
               114    * $80 = BEFEHL illegal
               115    * $81 = NAME0 = 0 (Präfix leer)
               116    * $82 = NAME1 = 0 (Name 1 leer)
               117    * $83 = NAME2 = 0 (Rename leer)
               118    * $84 = BLEN  = 0 (Bsave/Bload)
               119    * $85 = FLEN  > BLEN (Bload)
               120    *
               121    *  0 = Reboot ohne Parameter
               122    *  1 = Online ohne Parameter
               123    *  2 = Prefix NAME0
               124    *  3 = Cat    NAME1 oder NAME0
               125    *  4 = Create NAME1
               126    *  5 = Delete NAME1
               127    *  6 = Lock   NAME1
               128    *  7 = Unlock NAME1
               129    *  8 = Rename NAME1 + NAME2
               130    *  9 = Bsave  NAME1 + BANF + BLEN
               131    * 10 = Bload  NAME1 + BANF + BLEN
               132    *-------------------------------
46CE: 8D 8E 47 133    ENTRY    STA  ASAV
46D1: 8E 8F 47 134             STX  XSAV
46D4: 8C 90 47 135             STY  YSAV
46D7: A5 00    136             LDA  IND
46D9: 8D 8C 47 137             STA  INDSAV
46DC: A5 01    138             LDA  IND+1
46DE: 8D 8D 47 139             STA  INDSAV+1
46E1: A9 00    140             LDA  #0
46E3: 8D CC 46 141             STA  ERRMLI
46E6: 8D CD 46 142             STA  ERRALL
46E9: AD CB 46 143             LDA  BEFEHL
46EC: C9 0B    144             CMP  #10+1       ;0-10?
46EE: 90 08    145             BCC  ENTRY1
46F0: A9 80    146             LDA  #$80
46F2: 8D CD 46 147             STA  ERRALL
46F5: 4C 20 47 148             JMP  EXIT
46F8: 0A       149    ENTRY1   ASL              ;x 2
46F9: AA       150             TAX
46FA: BD 0A 47 151             LDA  JMPTAB,X    ;LL
46FD: 8D 08 47 152             STA  ENTRY2+1
4700: E8       153             INX
4701: BD 0A 47 154             LDA  JMPTAB,X    ;HH
4704: 8D 09 47 155             STA  ENTRY2+2
4707: 4C FF FF 156    ENTRY2   JMP  $FFFF       ;gepokt
               157    *
470A: 5D 49    158    JMPTAB   DA   REBOOT      ;0
470C: 60 49    159             DA   ONLINE      ;1
470E: F8 49    160             DA   PREFIX      ;2
```

```
471Ø: 37 4A     161          DA   CAT        ;3
4712: 38 4B     162          DA   CREATE     ;4
4714: 7B 4B     163          DA   DELETE     ;5
4716: 94 4B     164          DA   LOCKUNL    ;6
4718: 94 4B     165          DA   LOCKUNL    ;7
471A: F1 4B     166          DA   RENAME     ;8
471C: 12 4C     167          DA   BSAVE      ;9
471E: A7 4C     168          DA   BLOAD      ;1Ø
                169  *--------------------------------
                17Ø  *
                171  * ERRALL wird nicht angezeigt!
                172  *
472Ø: AD CC 46  173  EXIT     LDA  ERRMLI
4723: FØ 53     174          BEQ  EXITØ      ;okay
                175  *
                176  * MLI-Fehler aus Tabelle anzeigen
                177  *
4725: 2Ø 4D 49  178          JSR  CROUT
4728: A9 91     179          LDA  #<ERROR1
472A: 85 ØØ     18Ø          STA  IND
472C: A9 47     181          LDA  #>ERROR1
472E: 85 Ø1     182          STA  IND+1
473Ø: AØ ØØ     183          LDY  #Ø         ;Y=Ø
4732: B1 ØØ     184  EXIT1    LDA  (IND),Y    ;Loop
4734: FØ 1D     185          BEQ  EXIT4      ;Ø-Marke
4736: CD CC 46  186          CMP  ERRMLI
4739: FØ Ø8     187          BEQ  EXIT2      ;paßt!
473B: E6 ØØ     188          INC  IND
473D: DØ F3     189          BNE  EXIT1
473F: E6 Ø1     19Ø          INC  IND+1
4741: DØ EF     191          BNE  EXIT1
                192  *
                193  * Gefundenen Fehler ausgeben
                194  *
4743: C8        195  EXIT2    INY             ;bis zur
4744: B1 ØØ     196          LDA  (IND),Y    ;Ø-Marke
4746: DØ FB     197          BNE  EXIT2
4748: C8        198          INY
4749: B1 ØØ     199  EXIT3    LDA  (IND),Y
474B: FØ 2B     2ØØ          BEQ  EXITØ
474D: 2Ø 49 49  2Ø1          JSR  OUTDO
475Ø: C8        2Ø2          INY
4751: DØ F6     2Ø3          BNE  EXIT3      ;stets
                2Ø4  *
                2Ø5  * Noch eine Fehlermeldung?
                2Ø6  *
4753: E6 ØØ     2Ø7  EXIT4    INC  IND
4755: DØ Ø2     2Ø8          BNE  EXIT5
4757: E6 Ø1     2Ø9          INC  IND+1
4759: B1 ØØ     21Ø  EXIT5    LDA  (IND),Y    ;über
475B: DØ F6     211          BNE  EXIT4      ;Ascii
                212  *
475D: E6 ØØ     213          INC  IND        ;über
475F: DØ Ø2     214          BNE  EXIT6      ;Ø-Marke
4761: E6 Ø1     215          INC  IND+1
4763: B1 ØØ     216  EXIT6    LDA  (IND),Y    ;End-
4765: DØ CB     217          BNE  EXIT1      ;marke?
                218  *
                219  * Für sonstige Fehler nur MLI-Nr.
                22Ø  *
4767: B9 2D 48  221  EXIT7    LDA  MLIERROR,Y
476A: FØ Ø6     222          BEQ  EXIT8
476C: 2Ø 49 49  223          JSR  OUTDO
476F: C8        224          INY
477Ø: DØ F5     225          BNE  EXIT7
4772: AD CC 46  226  EXIT8    LDA  ERRMLI
4775: 2Ø 5A 49  227          JSR  HEXOUT
                228  *
4778: AD 8C 47  229  EXITØ    LDA  INDSAV
477B: 85 ØØ     23Ø          STA  IND
477D: AD 8D 47  231          LDA  INDSAV+1
478Ø: 85 Ø1     232          STA  IND+1
4782: AD 8E 47  233          LDA  ASAV
4785: AE 8F 47  234          LDX  XSAV
4788: AC 9Ø 47  235          LDY  YSAV
478B: 6Ø        236          RTS
                237  *
478C: ØØ ØØ     238  INDSAV   HEX  ØØØØ
478E: ØØ        239  ASAV     HEX  ØØ
478F: ØØ        24Ø  XSAV     HEX  ØØ
479Ø: ØØ        241  YSAV     HEX  ØØ
                242  *--------------------------------
4791: 27 28     243  ERROR1   HEX  2728       ;MLI-Nr.
4793: ØØ        244          HEX  ØØ         ;Ø-Marke
4794: CC E1 F5  245          ASC  "Laufwerksfehler"
47A3: ØØ        246          HEX  ØØ         ;Ø-Marke
                247  *
```

```
47A4: 2B 4E     248          HEX  2B4E
47A6: ØØ        249          HEX  ØØ
47A7: D3 E3 E8  25Ø          ASC  "Schreibgeschützt"
47B7: ØØ        251          HEX  ØØ
                252  *
47B8: 4Ø        253          HEX  4Ø
47B9: ØØ        254          HEX  ØØ
47BA: D5 EE E7  255          ASC  "Ungültiger Name"
47C9: ØØ        256          HEX  ØØ
                257  *
47CA: 44 45 46  258          HEX  444546
47CD: ØØ        259          HEX  ØØ
47CE: CE E9 E3  26Ø          ASC  "Nicht gefunden"
47DC: ØØ        261          HEX  ØØ
                262  *
47DD: 47 57     263          HEX  4757
47DF: ØØ        264          HEX  ØØ
47EØ: C4 EF FØ  265          ASC  "Doppelter Name"
47EE: ØØ        266          HEX  ØØ
                267  *
47EF: 48 49     268          HEX  4849
47F1: ØØ        269          HEX  ØØ
47F2: C4 E9 F3  27Ø          ASC  "Diskette oder"
                             ASC  " Inhaltsverzeichnis voll"
4817: ØØ        271          HEX  ØØ
                272  *
4818: 4A        273          HEX  4A
4819: ØØ        274          HEX  ØØ
481A: C6 E1 EC  275          ASC  "Falscher Dateityp"
482B: ØØ        276          HEX  ØØ
                277  *
482C: ØØ        278          HEX  ØØ         ;Endmark
                279  *
482D: CD CC C9  28Ø  MLIERROR ASC  "MLI-Fehler Nr. "
483C: ØØ        281          HEX  ØØ
                282  *================================
                283  *
                284  * Standard-MLI-Aufrufe
                285  * ====================
                286  *
                287  * Ø = Werte bleiben unverändert
                288  *     bzw. sind durch BPUF, DPUF,
                289  *     NAMEØ, NAME1 und NAME2
                29Ø  *     vordefiniert.
                291  * 1 = vor Aufruf selbst poken
                292  * 2 = nach Aufruf von MLI gepokt
                293  *
                294  *--------------------------------
483D: 2Ø ØØ BF  295  RB       JSR  MLI        ;RB=
484Ø: 65        296          HEX  65         ;Reboot
4841: 44 48     297          DA   RBCNT
4843: 6Ø        298          RTS
4844: Ø4        299  RBCNT    HEX  Ø4         ;Ø
                3ØØ  *--------------------------------
4845: 2Ø ØØ BF  3Ø1  CR       JSR  MLI        ;CR=
4848: CØ        3Ø2          HEX  CØ         ;Create
4849: 4C 48     3Ø3          DA   CRCNT
484B: 6Ø        3Ø4          RTS
484C: Ø7        3Ø5  CRCNT    HEX  Ø7         ;Ø
484D: 43 46     3Ø6  CRNAM    DA   NAME1      ;Ø
484F: C3        3Ø7  CRACC    HEX  C3         ;Ø (1)
485Ø: ØØ        3Ø8  CRTYP    HEX  ØØ         ;1
4851: ØØ ØØ     3Ø9  CRAUX    HEX  ØØØØ       ;1
4853: ØØ        31Ø  CRSTOR   HEX  ØØ         ;1
4854: ØØ ØØ     311  CRCDAT   HEX  ØØØØ       ;1
4856: ØØ ØØ     312  CRCUHR   HEX  ØØØØ       ;1
                313  *--------------------------------
4858: 2Ø ØØ BF  314  DE       JSR  MLI        ;DE=
485B: C1        315          HEX  C1         ;Destroy
485C: 5F 48     316          DA   DECNT
485E: 6Ø        317          RTS
485F: Ø1        318  DECNT    HEX  Ø1         ;Ø
486Ø: 43 46     319  DENAM    DA   NAME1      ;Ø
                32Ø  *--------------------------------
4862: 2Ø ØØ BF  321  RN       JSR  MLI        ;RN=
4865: C2        322          HEX  C2         ;Rename
4866: 69 48     323          DA   RNCNT
4868: 6Ø        324          RTS
4869: Ø2        325  RNCNT    HEX  Ø2         ;Ø
486A: 43 46     326  RNNAM1   DA   NAME1      ;Ø
486C: 84 46     327  RNNAM2   DA   NAME2      ;Ø
                328  *--------------------------------
486E: 2Ø ØØ BF  329  SF       JSR  MLI        ;SF=
4871: C3        33Ø          HEX  C3         ;Set
4872: 75 48     331          DA   SFCNT      ;File
4874: 6Ø        332          RTS             ;Info
4875: Ø7        333  SFCNT    HEX  Ø7         ;Ø
```

PEEKER Börse

Gelegenheitsanzeigen

Sie können unter dieser Rubrik zu einem besonders günstigen Preis

- Ihre Hardware und Software verkaufen
- Ihre Hard- und Software suchen
- Kontakte knüpfen und vieles mehr

Musteranzeige privat

1 Druckzeile à 32 Buchstaben nur DM 5,50 zuzügl. ges. MwSt. Mindestens 2 Druckzeilen

Beispiel:

> Verkaufe neuwertigen Typenraddrucker mit Apple-Interface. Preis auf Anfrage. Tel. 007

nur DM 18,81 inkl. MwSt.

Musteranzeige gewerblich

1 Druckzeile à 32 Buchstaben nur DM 11,– zuzügl. ges. MwSt. Mindestens 2 Druckzeilen

Beispiel:

> Neu im Angebot: Professionelle, separate Tastatur für Apple II plus 16 Funktionstasten und separatem Ziffernblock.
> Fa. Keyboard & Co.

nur DM 18,70 inkl. MwSt.

PEEKER

Peeker-Börse

AUFTRAG FÜR KLEINANZEIGEN

Bitte veröffentlichen Sie in der nächsterreichbaren Ausgabe nachstehenden Text unter folgender Rubrik:

☐ suche Hardware ☐ suche Software ☐ Verschiedenes
☐ biete Hardware ☐ biete Software ☐ Chiffre

Bitte jeweils 32 Buchstaben pro Zeile – einschließlich Satzzeichen und Wortzwischenräume. Bitte Absender nicht vergessen. Mindestens 2 Zeilen. Chiffregebühr DM 6,– zuzügl. MwSt.

PEEKER

Produkt-Karte

Zu der im **Peeker**, Heft _____, Seite _____ erschienenen

Anzeige über _____

bitte ich um detaillierte Information.

Ich wünsche ☐ Prospekt, Datenblatt ☐ Preisliste ☐ schriftliches Angebot ☐ tel. Rückruf

Menge	Produkt und Bestellnummer	à DM	gesamt DM

gebe ich nebenstehende Bestellung unter Anerkennung Ihrer in der Anzeige genannten Liefer- und Zahlungsbedingungen auf.

Unterschrift (für Jugendliche unter 18 Jahren der Erziehungsberechtigte)

PEEKER

Umfrage-Karte

Mit dieser Umfrage-Karte können Sie Einfluß auf die redaktionelle Gestaltung des Peeker nehmen. Deshalb Umfrage-Karte möglichst bald zurücksenden. Einsendeschluß: 30. September 1986

Ich besitze einen

☐ Apple-II-Plus ☐ Apple-II-Plus-Kompatiblen
☐ Apple-II-e/c ☐ Apple-II-e/c-Kompatiblen
☐ Mac/Lisa ☐ Sonstigen _____

Ich interessiere mich für den

☐ Apple-II-Nachfolger ☐ Mac-Nachfolger
☐ IBM-PC ☐ Atari-ST

Ich bin

☐ mehr Anwender ☐ mehr Programmierer

Peeker-Börse

Vorname, Name

Firma

Straße

Wohnort

PLZ/Ort

Bitte veröffentlichen Sie den umstehenden Text von _____ Zeilen à _____ DM in der nächsterreichbaren Ausgabe vom **Peeker**

Bei Angeboten: Ich bestätige, daß ich alle Rechte an den angebotenen Sachen besitze

Datum Unterschrift

Produkt-Karte

Karte bitte vollständig ausfüllen

Vorname, Name

Firma

Straße

PLZ/Ort

Telefon mit Vorwahl

Anschrift der Firma angeben, bei der Sie bestellen bzw. von der Sie Informationen wünschen

Umfrage-Karte

Karte bitte vollständig ausfüllen

Vorname, Name

Firma

Straße

PLZ/Ort

Telefon mit Vorwahl

POSTKARTE

Peeker-Börse
Anzeigen-Service

Dr. Alfred Hüthig Verlag

Postfach 10 28 69

6900 Heidelberg 1

POSTKARTE

Inserent

Straße

PLZ/Ort

POSTKARTE

Peeker
Redaktion

Dr. Alfred Hüthig Verlag

Postfach 10 28 69

6900 Heidelberg 1

Produkt-Karte

Wünschen Sie weitere Informationen zu einer der im Heft erschienenen Anzeigen?

Nichts einfacher als das. Produkt-Karte ausfüllen, frankieren und an den Inserenten (nicht an die Peeker-Redaktion) senden.

Vorher aber nicht vergessen:
Kreuzen Sie an, welchen Informationswunsch Sie haben.

Damit erleichtern Sie dem Hersteller eine gezielte Beantwortung Ihrer Anfrage

Zum Schluß tragen Sie auf der Rückseite die genaue Anschrift des Inserenten und Ihren Absender ein.

PEEKER

```
4876: 43 46     334  SFNAM    DA   NAME1       ;Ø
4878: ØØ        335  SFACC    HEX  ØØ          ;1
4879: ØØ        336  SFTYP    HEX  ØØ          ;1
487A: ØØ ØØ     337  SFAUX    HEX  ØØØØ        ;1
487C: ØØ        338  SFNULL1  HEX  ØØ          ;Ø
487D: ØØ ØØ     339  SFNULL2  HEX  ØØØØ        ;Ø
487F: ØØ ØØ     34Ø  SFMDAT   HEX  ØØØØ        ;1
4881: ØØ ØØ     341  SFMUHR   HEX  ØØØØ        ;1
                342  *--------------------------------
4883: 2Ø ØØ BF  343  GF       JSR  MLI         ;GF=
4886: C4        344            HEX  C4          ;Get
4887: 8A 48     345            DA   GFCNT       ;File
4889: 6Ø        346            RTS              ;Info
488A: ØA        347  GFCNT    HEX  ØA          ;Ø
488B: 43 46     348  GFNAM    DA   NAME1       ;Ø
488D: ØØ        349  GFACC    HEX  ØØ          ;2
488E: ØØ        35Ø  GFTYP    HEX  ØØ          ;2
488F: ØØ ØØ     351  GFAUX    HEX  ØØØØ        ;2
4891: ØØ        352  GFSTOR   HEX  ØØ          ;2
4892: ØØ ØØ     353  GFBLKS   HEX  ØØØØ        ;2
4894: ØØ ØØ     354  GFMDAT   HEX  ØØWØ        ;2
4896: ØØ ØØ     355  GFMUHR   HEX  ØØØØ        ;2
4898: ØØ ØØ     356  GFCDAT   HEX  ØØØØ        ;2
489A: ØØ ØØ     357  GFCUHR   HEX  ØØØØ        ;2
                358  *--------------------------------
489C: 2Ø ØØ BF  359  OL       JSR  MLI         ;OL=
489F: C5        36Ø            HEX  C5          ;Online
48AØ: A3 48     361            DA   ONCNT
48A2: 6Ø        362            RTS
48A3: Ø2        363  ONCNT    HEX  Ø2          ;Ø
48A4: ØØ        364  ONUNIT   HEX  ØØ          ;Ø (1)
48A5: ØØ 44     365  ONDPUF   DA   DPUF        ;Ø
                366  *--------------------------------
48A7: 2Ø ØØ BF  367  SP       JSR  MLI         ;SP=
48AA: C6        368            HEX  C6          ;Set
48AB: AE 48     369            DA   SPCNT       ;Prefix
48AD: 6Ø        37Ø            RTS
48AE: Ø1        371  SPCNT    HEX  Ø1          ;Ø
48AF: ØØ 46     372  SPNAMEØ  DA   NAMEØ       ;Ø
                373  *--------------------------------
48B1: 2Ø ØØ BF  374  GP       JSR  MLI         ;GP=
48B4: C7        375            HEX  C7          ;Get
48B5: B8 48     376            DA   GPCNT       ;Prefix
48B7: 6Ø        377            RTS
48B8: Ø1        378  GPCNT    HEX  Ø1          ;Ø
48B9: ØØ 46     379  GPNAMEØ  DA   NAMEØ       ;Ø
                38Ø  *--------------------------------
48BB: 2Ø ØØ BF  381  OP       JSR  MLI         ;OP=
48BE: C8        382            HEX  C8          ;Open
48BF: C2 48     383            DA   OPCNT
48C1: 6Ø        384            RTS
48C2: Ø3        385  OPCNT    HEX  Ø3          ;Ø
48C3: 43 46     386  OPNAM    DA   NAME1       ;Ø
48C5: ØØ 4Ø      387  OPBPUF   DA   BPUF        ;Ø
48C7: ØØ        388  OPREF    HEX  ØØ          ;2
                389  *--------------------------------
48C8: 2Ø ØØ BF  39Ø  NL       JSR  MLI         ;NL=
48CB: C9        391            HEX  C9          ;Newline
48CC: CF 48     392            DA   NLCNT
48CE: 6Ø        393            RTS
48CF: Ø3        394  NLCNT    HEX  Ø3          ;Ø
48DØ: ØØ        395  NLREF    HEX  ØØ          ;1
48D1: ØØ        396  NLENAB   HEX  ØØ          ;Ø (1)
48D2: ØØ        397  NLCHAR   HEX  ØØ          ;Ø (1)
                398  *--------------------------------
48D3: 2Ø ØØ BF  399  RD       JSR  MLI         ;RD=
48D6: CA        4ØØ            HEX  CA          ;Read
48D7: DA 48     4Ø1            DA   RDCNT
48D9: 6Ø        4Ø2            RTS
48DA: Ø4        4Ø3  RDCNT    HEX  Ø4          ;Ø
48DB: ØØ        4Ø4  RDREF    HEX  ØØ          ;1
48DC: ØØ ØØ     4Ø5  RDDPUF   HEX  ØØØØ        ;1
48DE: ØØ ØØ     4Ø6  RDREQ    HEX  ØØØØ        ;1
48EØ: ØØ ØØ     4Ø7  RDTRA    HEX  ØØØØ        ;2
                4Ø8  *--------------------------------
48E2: 2Ø ØØ BF  4Ø9  WR       JSR  MLI         ;WR=
48E5: CB        41Ø            HEX  CB          ;Write
48E6: E9 48     411            DA   WRCNT
48E8: 6Ø        412            RTS
48E9: Ø4        413  WRCNT    HEX  Ø4          ;Ø
48EA: ØØ        414  WRREF    HEX  ØØ          ;1
48EB: ØØ ØØ     415  WRDPUF   HEX  ØØØØ        ;1
48ED: ØØ ØØ     416  WRREQ    HEX  ØØØØ        ;1
48EF: ØØ ØØ     417  WRTRA    HEX  ØØØØ        ;2
                418  *--------------------------------
48F1: 2Ø ØØ BF  419  CL       JSR  MLI         ;CL=
48F4: CC        42Ø            HEX  CC          ;Close
```

```
48F5: F8 48     421            DA   CLCNT
48F7: 6Ø        422            RTS
48F8: Ø1        423  CLCNT    HEX  Ø1          ;Ø
48F9: ØØ        424  CLREF    HEX  ØØ          ;Ø (1)
                425  *--------------------------------
48FA: 2Ø ØØ BF  426  FL       JSR  MLI         ;FL=       nicht
48FD: CD        427            HEX  CD          ;Flush     benutzt
48FE: Ø1 49     428            DA   FLCNT
49ØØ: 6Ø        429            RTS
49Ø1: Ø1        43Ø  FLCNT    HEX  Ø1          ;Ø
49Ø2: ØØ        431  FLREF    HEX  ØØ          ;Ø (1)
                432  *--------------------------------
49Ø3: 2Ø ØØ BF  433  SM       JSR  MLI         ;SM=       nicht
49Ø6: CE        434            HEX  CE          ;Set       benutzt
49Ø7: ØA 49     435            DA   SMCNT       ;Mark
49Ø9: 6Ø        436            RTS
49ØA: Ø2        437  SMCNT    HEX  Ø2          ;Ø
49ØB: ØØ        438  SMREF    HEX  ØØ          ;1
49ØC: ØØ ØØ ØØ  439  SMPOS    HEX  ØØØØØØ      ;1
                44Ø  *--------------------------------
49ØF: 2Ø ØØ BF  441  GM       JSR  MLI         ;GM=       nicht
4912: CF        442            HEX  CF          ;Get       benutzt
4913: 16 49     443            DA   GMCNT       ;Mark
4915: 6Ø        444            RTS
4916: Ø2        445  GMCNT    HEX  Ø2          ;Ø
4917: ØØ        446  GMREF    HEX  ØØ          ;1
4918: ØØ ØØ ØØ  447  GMPOS    HEX  ØØØØØØ      ;2
                448  *--------------------------------
491B: 2Ø ØØ BF  449  SE       JSR  MLI         ;SE=       nicht
491E: DØ        45Ø            HEX  DØ          ;Set EOF   benutzt
491F: 22 49     451            DA   SECNT
4921: 6Ø        452            RTS
4922: Ø2        453  SECNT    HEX  Ø2          ;Ø
4923: ØØ        454  SEREF    HEX  ØØ          ;1
4924: ØØ ØØ ØØ  455  SEEOF    HEX  ØØØØØØ      ;1
                456  *--------------------------------
4927: 2Ø ØØ BF  457  GE       JSR  MLI         ;GE=
492A: D1        458            HEX  D1          ;Get EOF
492B: 2E 49     459            DA   GECNT
492D: 6Ø        46Ø            RTS
492E: Ø2        461  GECNT    HEX  Ø2          ;Ø
492F: ØØ        462  GEREF    HEX  ØØ          ;1
493Ø: ØØ ØØ ØØ  463  GEEOF    HEX  ØØØØØØ      ;2
                464  *--------------------------------
4933: 2Ø ØØ BF  465  SB       JSR  MLI         ;SB=       nicht
4936: D2        466            HEX  D2          ;Set       benutzt
4937: 3A 49     467            DA   SBCNT       ;Buffer
4939: 6Ø        468            RTS
493A: Ø2        469  SBCNT    HEX  Ø2          ;Ø
493B: ØØ        47Ø  SBREF    HEX  ØØ          ;1
493C: ØØ 4Ø      471  SBBPUF   DA   BPUF        ;Ø
                472  *--------------------------------
493E: 2Ø ØØ BF  473  GB       JSR  MLI         ;GB=       nicht
4941: D3        474            HEX  D3          ;Get       benutzt
4942: 45 49     475            DA   GBCNT       ;Buffer
4944: 6Ø        476            RTS
4945: Ø2        477  GBCNT    HEX  Ø2          ;Ø
4946: ØØ        478  GBREF    HEX  ØØ          ;1
4947: ØØ ØØ     479  GBBPUF   HEX  ØØØØ        ;2
                48Ø  *
                481  *================================
                482  *
                483  * Ausgaberoutinen
                484  * ================
                485  *
4949: Ø9 8Ø      486  OUTDO    ORA  #$8Ø
494B: DØ Ø6      487            BNE  OUTCHAR
494D: A9 8D      488  CROUT    LDA  #$8D        ;Return
494F: DØ Ø2      489            BNE  OUTCHAR
4951: A9 AØ      49Ø  OUTSPC   LDA  #$AØ        ;Space
4953: 4C ED FD   491  OUTCHAR  JMP  COUT        ;A=Char
4956: 2Ø 5A 49   492  HEXAX    JSR  HEXOUT      ;A=HH
4959: 8A        493            TXA              ;X=LL
495A: 4C DA FD   494  HEXOUT   JMP  PRBYTE      ;A=Hex
                495  *================================
                496  *
                497  * Ø = REBOOT - ohne Parameter
                498  * ==========================
                499  *
                5ØØ  * Startet anderen SYS-File, den
                5Ø1  * man über Tastatur eingibt.
                5Ø2  *
495D: 4C 3D 48  5Ø3  REBOOT   JMP  RB
                5Ø4  *--------------------------------
                5Ø5  *
                5Ø6  * 1 = ONLINE - ohne Parameter
                5Ø7  * ==========================
```

```
                 508  *
                 509  * Zeigt an:    Ss,Dd /Volume1
                 510  *              Ss,Dd /Volume2 usw.
                 511  * zudem:       /Aktuelles Präfix
                 512  *
496Ø: A9 ØØ      513  ONLINE   LDA  #<DPUF
4962: 85 ØØ      514           STA  IND
4964: A9 44      515           LDA  #>DPUF
4966: 85 Ø1      516           STA  IND+1
4968: AØ ØØ      517           LDY  #Ø
496A: 98         518           TYA
496B: 91 ØØ      519  ONLINE1  STA  (IND),Y     ;Puffer
496D: C8         52Ø           INY              ;löschen
496E: DØ FB      521           BNE  ONLINE1
497Ø: 2Ø 9C 48   522           JSR  OL
4973: 9Ø 1C      523           BCC  ONLPØ
4975: 8D CC 46   524           STA  ERRMLI
4978: 4C 2Ø 47   525  ONLINE2  JMP  EXIT
                 526  *
                 527  * Zusätzlichh ggf. Präfix anzeigen
                 528  *
497B: AE ØØ 46   529  ONLINE3  LDX  NAMEØ       ;leer?
497E: FØ F8      53Ø           BEQ  ONLINE2
498Ø: 2Ø 4D 49   531           JSR  CROUT
4983: AØ ØØ      532           LDY  #Ø
4985: C8         533  ONLINE4  INY
4986: B9 ØØ 46   534           LDA  NAMEØ,Y
4989: 2Ø 49 49   535           JSR  OUTDO
498C: CA         536           DEX
498D: DØ F6      537           BNE  ONLINE4
498F: FØ E7      538           BEQ  ONLINE2
                 539  *
4991: AØ ØØ      54Ø  ONLPØ    LDY  #Ø           ;Loop
4993: B1 ØØ      541           LDA  (IND),Y
4995: DØ Ø5      542           BNE  ONLP1
4997: C8         543           INY
4998: B1 ØØ      544           LDA  (IND),Y
499A: FØ DF      545           BEQ  ONLINE3      ;Präfix
                 546  *
499C: 2Ø 4D 49   547  ONLP1    JSR  CROUT
499F: AØ ØØ      548           LDY  #Ø
49A1: A9 D3      549           LDA  #"S"         ;Slot-
49A3: 2Ø 49 49   55Ø           JSR  OUTDO        ,anzeige
49A6: B1 ØØ      551           LDA  (IND),Y
49A8: 29 7Ø      552           AND  #%Ø111ØØØØ
49AA: 4A         553           LSR
49AB: 4A         554           LSR
49AC: 4A         555           LSR
49AD: 4A         556           LSR
49AE: Ø9 BØ      557           ORA  #$BØ         ;Ø
49BØ: 2Ø 49 49   558           JSR  OUTDO        ;Slot
49B3: A9 AC      559           LDA  #","
49B5: 2Ø 49 49   56Ø           JSR  OUTDO
49B8: B1 ØØ      561           LDA  (IND),Y
49BA: 29 ØF      562           AND  #%ØØØØ1111
49BC: AA         563           TAX               ;Länge
                 564  *
49BD: A9 C4      565           LDA  #"D"         ;Drive-
49BF: 2Ø 49 49   566           JSR  OUTDO        ;anzeige
49C2: B1 ØØ      567           LDA  (IND),Y
49C4: 1Ø Ø4      568           BPL  ONLP2
49C6: A9 B2      569           LDA  #"2"         ;D2
49C8: DØ Ø2      57Ø           BNE  ONLP3
49CA: A9 B1      571  ONLP2    LDA  #"1"         ;D1
49CC: 2Ø 49 49   572  ONLP3    JSR  OUTDO
49CF: 2Ø 51 49   573           JSR  OUTSPC
                 574  *
49D2: EØ ØØ      575           CPX  #Ø           ;Len=Ø?
49D4: FØ 1Ø      576           BEQ  ONLP5        ;Fehler!
49D6: A9 AF      577           LDA  #"/"
49D8: 2Ø 49 49   578           JSR  OUTDO
49DB: C8         579  ONLP4    INY               ;Namens-
49DC: B1 ØØ      58Ø           LDA  (IND),Y      ;anzeige
49DE: 2Ø 49 49   581           JSR  OUTDO
49E1: CA         582           DEX
49E2: DØ F7      583           BNE  ONLP4
49E4: FØ Ø5      584           BEQ  ONLP6
                 585  *
49E6: A9 BF      586  ONLP5    LDA  #"?"         ;leer!
49E8: 2Ø 49 49   587           JSR  OUTDO
                 588  *
49EB: 18         589  ONLP6    CLC               ;näch-
49EC: A5 ØØ      59Ø           LDA  IND          ;ster
49EE: 69 1Ø      591           ADC  #16          ,Name
49FØ: 85 ØØ      592           STA  IND
49F2: 9Ø 9D      593           BCC  ONLPØ
49F4: E6 Ø1      594           INC  IND+1
49F6: DØ 99      595           BNE  ONLPØ
                 596  *-----------------------------
                 597  *
                 598  * 2 = PREFIX mit Parameter:
                 599  * ========================
                 6ØØ  *
                 6Ø1  * (a) Für GETPREFIX das NAMEØ-
                 6Ø2  *     Längebyte auf Ø setzen.
                 6Ø3  *     Ermittelt und zeigt Präfix
                 6Ø4  *     an, bei Nullpräfix "?".
                 6Ø5  * (b) Für SETPREFIX in NAMEØ
                 6Ø6  *     Längebyte + neues Präfix
                 6Ø7  *     eintragen. Also:
                 6Ø8  *
                 6Ø9  * NAMEØ = Ø       -> GETPREFIX
                 61Ø  * NAMEØ = belegt -> SETPREFIX
                 611  *
                 612  * Das in NAMEØ abgelegte Präfix
                 613  * ist mit dem ProDOS-immanenten
                 614  * Präfix identisch, wenn bei
                 615  * Programmstart zunächst PREFIX
                 616  * bzw. Befehl 2 aufgerufen wird.
                 617  *
49F8: AD ØØ 46   618  PREFIX   LDA  NAMEØ       ;leer?
49FB: FØ 17      619           BEQ  GETFIX
                 62Ø  *
49FD: 2Ø A7 48   621  SETFIX   JSR  SP
4AØØ: 9Ø Ø8      622           BCC  PREFIX3
                 623  *
4AØ2: 8D CC 46   624  PREFIX1  STA  ERRMLI
4AØ5: A9 ØØ      625  PREFIX2  LDA  #Ø
4AØ7: 8D ØØ 46   626           STA  NAMEØ       ;Präf=Ø!
4AØA: AD ØØ 46   627  PREFIX3  LDA  NAMEØ       ;>64 wg.
                 628  *
                 629  * Achtung: Im Extremfall kann
                 63Ø  * GETFIX ein Präfix mit mehr als
                 631  * 64 Zeichen eintragen.
                 632  * Bei SETFIX darf man jedoch
                 633  * nie mehr als 64 Ascii verwenden
                 634  *
4AØD: C9 41      635           CMP  #65         ;>=64?
4AØF: BØ F4      636           BCS  PREFIX2     ;löschen
4A11: 4C 2Ø 47   637           JMP  EXIT
                 638  *
4A14: 2Ø B1 48   639  GETFIX   JSR  GP
4A17: BØ E9      64Ø           BCS  PREFIX1
                 641  *
                 642  * Ermitteltes Präfix anzeigen
                 643  *
4A19: 2Ø 4D 49   644           JSR  CROUT
4A1C: AØ ØØ      645           LDY  #Ø
4A1E: AE ØØ 46   646           LDX  NAMEØ       ;Länge
4A21: DØ Ø8      647           BNE  GETFIX1
4A23: A9 BF      648           LDA  #"?"        ;leer
4A25: 2Ø 49 49   649           JSR  OUTDO
4A28: 4C ØA 4A   65Ø           JMP  PREFIX3
4A2B: C8         651  GETFIX1  INY
4A2C: B9 ØØ 46   652           LDA  NAMEØ,Y
4A2F: 2Ø 49 49   653           JSR  OUTDO
4A32: CA         654           DEX
4A33: DØ F6      655           BNE  GETFIX1
4A35: FØ D3      656           BEQ  PREFIX3
                 657  *-----------------------------
                 658  *
                 659  * 3 = CAT mit Parameter:
                 66Ø  * ========================
                 661  *
                 662  * a) Wenn man vorher das Länge-
                 663  *    Byte von NAME1 auf Ø setzt,
                 664  *    wird NAMEØ = Default-Präfix
                 665  *    verwendet, das aber zuvor
                 666  *    gesetzt worden sein muß.
                 667  * b) Wenn man indes in NAME1 den
                 668  *    gewünschten Directory-Namen
                 669  *    einträgt, wird dieser ver-
                 67Ø  *    verwendet.  Also:
                 671  *
                 672  * a) NAME1 = Ø, dann NAMEØ
                 673  * b) NAME1 > Ø, dann NAME1
                 674  *
                 675  * Catalog-Anzeige in der Form an:
                 676  *
                 677  *  *      A    ØØØØ     NAME
                 678  *  Lock   Typ  Blocks   Dateiname
                 679  *  Unlock       in Hex
                 68Ø  *
4A37: AE 43 46   681  CAT      LDX  NAME1       ;leer?
```

Checkmate Technology

™ **Speichererweiterungen**
mit Zukunft
für Apple IIe und Apple IIc

Multiram Leistungsmerkmale:
- 16bit CPU Port
- 65C816 Coprozessor, der den Multiram Speicher linear addressieren kann (sofort lieferbar)
- inclusive Apple Works Memory Expander (endlich genug Speicherplatz für Appleworks)
- inkl. Ramdisk Software für DOS und ProDos
- Ramdisk für Pascal und CP/M optional

speziell beim Apple IIe:
- auf einer Karte bis max. 1MB erweiterbar mit Huckepackkarte auf 1,7 MB erweiterbar
- kommt in Auxiliary Slot (3)
- ersetzt erweiterte 80 Zeichen Karte
- RGB Farboption lieferbar

speziell beim Apple IIc:
- bis 512k erweiterbar
- ohne externes Laufwerk echt portabel

pandasoft Dr.-Ing. Eden

Kataloganforderung und Bestellung: Tel.: 030/31 04 23 · Telex 185 859
Uhlandstraße 195 · D-1000 Berlin 12

kyan PASCAL

Pascal Compiler für Apple II (+,e,c)
- erzeugt 6502 Assembler-Code
- benötigt keine Z-80 Karte
- läuft unter Prodos
- integrierter Assembler
- inclusive Editor (full screen)
- mehrfach in Peeker getestet
- neu: Kix eine Uni™ ähnliche Betriebssystem Umgebung

Kennenlernaktion bis 30. Juni:
Kyan 2.0 inkl. KIX: DM 198,–

nach Ablauf der Kennenlernaktion:
Kyan 2.0 alleine DM 198,00
Kyan 2.0 inkl. Kix: DM 278,00

Unix ist ein eingetragenes Warenzeichen der Bell Laboratories

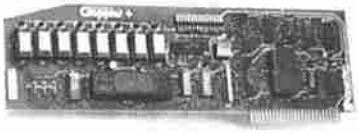
pandasoft Dr.-Ing. Eden

Kataloganforderung und Bestellung: Tel.: 030/31 04 23 · Telex 185 859
Uhlandstraße 195 · D-1000 Berlin 12

Orange Printer Interface — **Druckerinterfaces** für Apple II+/e/c/III Interfaces auf dem **neuesten Stand der Technik. Kompatibel** mit allen gängigen Druckern wie: APPLE, EPSON, STAR, NEC, OKIDATA usw. Passende Treibersoftware wird über Dip-Switch ausgewählt.

Grappler + Printer Interface — **Grafikfähiges Druckerinterface** das keine Wünsche mehr offen läßt. Über **2 Dutzend Kommandos** ermöglichen die volle Kontrolle über alle Möglichkeiten Ihres Druckers. Jetzt auch mit **IIe Features: Double Hires Grafics** und **80 Zeichen Dump** mittels Druckerpuffer nachrüstbar über Bufferboard.

Buffer Board Grappler + — Besitzt alle Vorzüge des Grappler +, hat aber zusätzlich einen integrierten **16 K Druckpuffer,** der auf **32 oder 64 K aufrüstbar** ist.



Serial Grappler Printer Interface — **Serielles Druckerinterface** speziell für den **Apple Imagewriter.**

HOTLINK — Seriell-nach-Parallel-Wandler für den IIc im Kabel integriert.

GRAPPLER C — wie Hotlink, jedoch zusätzlich Imagewriter Emulation und Grafik Software-Diskette.

pandasoft Dr.-Ing. Eden

Kataloganforderung und Bestellung: Tel.: 030/31 04 23 · Telex 185 859
Uhlandstraße 195 · D-1000 Berlin 12

Sie haben einen Apple ...

wir haben die Software ... | und die Hardware ...

wir haben die Bücher ... | und die Zeitschriften *...

*Fordern Sie unseren Gratiskatalog an!

ALLES FÜR DEN APPLE II+, IIe, IIc UND MACINTOSH

pandasoft Dr.-Ing. Eden

Uhlandstraße 195 · D-1000 Berlin 12
Tel.: 030/31 04 23 · Telex 185 859

Ich besitze einen Apple II+, IIe, IIc Bitte schicken Sie mir Ihren 168 Seiten Apple II Katalog
Name Adresse

```
4A3A: DØ 1B      682            BNE  CAT1
4A3C: AE ØØ 46   683            LDX  NAMEØ       ;leer?
4A3F: DØ Ø7      684            BNE  CATMOVE1
4A41: A9 81      685            LDA  #$81
4A43: 8D CD 46   686            STA  ERRALL
4A46: DØ 26      687            BNE  CAT4
                 688  *
                 689  * NAMEØ nach NAME1 kopieren
                 69Ø  *
4A48: 8E 43 46   691  CATMOVE1 STX  NAME1       ;Länge
4A4B: AØ ØØ      692            LDY  #Ø
4A4D: C8         693  CATMOVE2 INY
4A4E: B9 ØØ 46   694            LDA  NAMEØ,Y
4A51: 99 43 46   695            STA  NAME1,Y
4A54: CA         696            DEX
4A55: DØ F6      697            BNE  CATMOVE2
                 698  *
4A57: 2Ø 83 48   699  CAT1     JSR  GF
4A5A: A9 4A      7ØØ            LDA  #$4A        ;illegal
4A5C: AE 8E 48   7Ø1            LDX  GFTYP       ;Type
4A5F: EØ ØF      7Ø2            CPX  #$ØF        ;Dir?
4A61: DØ Ø5      7Ø3            BNE  CAT2
4A63: 2Ø BB 48   7Ø4            JSR  OP
4A66: 9Ø Ø9      7Ø5            BCC  CAT5
                 7Ø6  *
4A68: 8D CC 46   7Ø7  CAT2     STA  ERRMLI
                 7Ø8  *
                 7Ø9  * Close wird trotz möglicher
                 71Ø  * Fehler immer versucht!
                 711  *
4A6B: 2Ø F1 48   712  CAT3     JSR  CL          ;Close
4A6E: 4C 2Ø 47   713  CAT4     JMP  EXIT
                 714  *
4A71: AD C7 48   715  CAT5     LDA  OPREF       ;Newline
4A74: 8D DØ 48   716            STA  NLREF
4A77: 2Ø C8 48   717            JSR  NL
4A7A: BØ EC      718            BCS  CAT2
                 719  *
                 72Ø  * Einen Directory-Block lesen
                 721  *
4A7C: AD C7 48   722  CAT6     LDA  OPREF       ;Dir-Blk
4A7F: 8D DB 48   723            STA  RDREF       ;lesen
4A82: A9 ØØ      724            LDA  #<DPUF
4A84: 8D DC 48   725            STA  RDDPUF
4A87: A9 44      726            LDA  #>DPUF
4A89: 8D DD 48   727            STA  RDDPUF+1
4A8C: A9 ØØ      728            LDA  #<$Ø2ØØ     ;$Ø2ØØ
4A8E: 8D DE 48   729            STA  RDREQ       ;=1 Blk.
4A91: A9 Ø2      73Ø            LDA  #>$Ø2ØØ
4A93: 8D DF 48   731            STA  RDREQ+1
4A96: 2Ø D3 48   732            JSR  RD
4A99: BØ CD      733            BCS  CAT2
4A9B: A9 Ø4      734            LDA  #<DPUF+4    ;Blk.+4
4A9D: 85 ØØ      735            STA  IND         ;Offset
4A9F: A9 44      736            LDA  #>DPUF
4AA1: 85 Ø1      737            STA  IND+1
                 738  *
                 739  * Kopf + Delete-Files ignorieren
                 74Ø  *
4AA3: AØ ØØ      741  CAT7     LDY  #Ø          ;Eintrag
4AA5: B1 ØØ      742            LDA  (IND),Y     ;1.Byte
4AA7: FØ 5F      743            BEQ  CAT15       ;deleted
                 744  *
4AA9: 29 FØ      745            AND  #$FØ        ;Vol/Dir
4AAB: C9 EØ      746            CMP  #$EØ        ;$E/$F
4AAD: BØ 59      747            BCS  CAT15       ;Kopf!
                 748  *
                 749  * Eintrag anzeigen
                 75Ø  *
4AAF: 2Ø 4D 49   751            JSR  CROUT       ;Return
4AB2: AØ 1E      752            LDY  #$1E        ;Access
4AB4: B1 ØØ      753            LDA  (IND),Y
4AB6: A2 AØ      754            LDX  #$AØ        ;Space
4AB8: 29 C2      755            AND  #%11ØØØØ1Ø
4ABA: C9 C2      756            CMP  #%11ØØØØ1Ø  ;unlock?
4ABC: FØ Ø2      757            BEQ  CAT8
4ABE: A2 AA      758            LDX  #"*"        ;locked!
4ACØ: 8A         759  CAT8     TXA
4AC1: 2Ø 49 49   76Ø            JSR  OUTDO
                 761  *
4AC4: AØ 1Ø      762            LDY  #$1Ø        ;Typ-
4AC6: B1 ØØ      763            LDA  (IND),Y     ;Offset
4AC8: A2 Ø4      764            LDX  #4          ;5 Typen
4ACA: DD D6 4A   765  CAT9     CMP  CAT1Ø,X
4ACD: FØ 11      766            BEQ  CAT12
4ACF: CA         767            DEX
4ADØ: 1Ø F8      768            BPL  CAT9
```

```
4AD2: A9 BF      769            LDA  #"?"        ;restl.
4AD4: DØ ØD      77Ø            BNE  CAT13
                 771  *
                 772  * TXT, BIN, DIR, A=BAS, SYS
                 773  *
4AD6: Ø4 Ø6 ØF   774  CAT1Ø    HEX  Ø4Ø6ØFFCFF
4ADB: D4 C2 C4   775  CAT11    ASC  "TBDAS"
                 776  *
4AEØ: BD DB 4A   777  CAT12    LDA  CAT11,X
4AE3: 2Ø 49 49   778  CAT13    JSR  OUTDO
4AE6: 2Ø 51 49   779            JSR  OUTSPC
                 78Ø  *
4AE9: AØ 13      781            LDY  #$13        ;Blk-LL
4AEB: B1 ØØ      782            LDA  (IND),Y
4AED: AA         783            TAX              ;X=LL
4AEE: AØ 14      784            LDY  #$14        ;Blk-HH
4AFØ: B1 ØØ      785            LDA  (IND),Y     ;A=HH
4AF2: 2Ø 56 49   786            JSR  HEXAX
4AF5: 2Ø 51 49   787            JSR  OUTSPC
                 788  *
4AF8: AØ ØØ      789            LDY  #Ø          ;Namens-
4AFA: B1 ØØ      79Ø            LDA  (IND),Y     ;anzeige
4AFC: 29 ØF      791            AND  #$ØF        ;Länge
4AFE: AA         792            TAX
4AFF: C8         793  CAT14    INY
4BØØ: B1 ØØ      794            LDA  (IND),Y
4BØ2: 2Ø 49 49   795            JSR  OUTDO
4BØ5: CA         796            DEX
4BØ6: DØ F7      797            BNE  CAT14
                 798  *
4BØ8: 18         799  CAT15    CLC
4BØ9: A5 ØØ      8ØØ            LDA  IND         ;Eintrag
4BØB: 69 27      8Ø1            ADC  #39         ;Offset
4BØD: 85 ØØ      8Ø2            STA  IND
4BØF: A5 Ø1      8Ø3            LDA  IND+1
4B11: 69 ØØ      8Ø4            ADC  #Ø
4B13: 85 Ø1      8Ø5            STA  IND+1
                 8Ø6  *
                 8Ø7  * $FF-Abbruch funktioniert nur,
                 8Ø8  * wenn Puffer auf Seitengrenze
                 8Ø9  *
4B15: A5 ØØ      81Ø            LDA  IND
4B17: C9 FF      811            CMP  #$FF        ;$Ø1FF
4B19: FØ Ø3      812            BEQ  CAT16
4B1B: 4C A3 4A   813            JMP  CAT7
                 814  *
                 815  * Noch ein Block? (Ptr <> Ø)
                 816  *
4B1E: A9 Ø2      817  CAT16    LDA  #<DPUF+2    ;Vorw.
4B2Ø: 85 ØØ      818            STA  IND         ;Zeiger
4B22: A9 44      819            LDA  #>DPUF
4B24: 85 Ø1      82Ø            STA  IND+1
4B26: AØ ØØ      821            LDY  #Ø
4B28: 18         822            CLC
4B29: B1 ØØ      823            LDA  (IND),Y
4B2B: C8         824            INY
4B2C: 71 ØØ      825            ADC  (IND),Y
4B2E: DØ Ø3      826            BNE  CAT17       ;ja
4B3Ø: 4C 6B 4A   827            JMP  CAT3        ;nein
4B33: 4C 7C 4A   828  CAT17    JMP  CAT6
                 829  *-----------------------------
                 83Ø  *
                 831  * 4 = CREATE mit Parameter:
                 832  * ==========================
                 833  *
                 834  * Directory-Name in NAME1 ablegen
                 835  * Es wird also nur ein DIR-File
                 836  * erzeugt (ggf. unten ändern).
                 837  *
4B36: ØF         838  CRTYPØ   HEX  ØF           ;Dir!
4B37: ØD         839  CRSTORØ  HEX  ØD           ;Dir!
                 84Ø  *
4B38: AD ØØ 46   841  CREATE   LDA  NAMEØ        ;leer?
4B3B: FØ Ø7      842            BEQ  CREATE1
4B3D: AD 43 46   843            LDA  NAME1       ;leer?
4B4Ø: FØ Ø6      844            BEQ  CREATE2
4B42: DØ ØC      845            BNE  CREATE5
4B44: A9 81      846  CREATE1   LDA  #$81
4B46: DØ Ø2      847            BNE  CREATE3
4B48: A9 82      848  CREATE2   LDA  #$82
4B4A: 8D CD 46   849  CREATE3   STA  ERRALL
4B4D: 4C 2Ø 47   85Ø  CREATE4   JMP  EXIT
                 851  *
4B5Ø: AD 36 4B   852  CREATE5   LDA  CRTYPØ      ;DIR!
4B53: 8D 5Ø 48   853            STA  CRTYP       ;Type
4B56: A9 ØØ      854            LDA  #$ØØ
4B58: 8D 51 48   855            STA  CRAUX       ;Auxtype
```

```
4B5B: A9 ØØ      856            LDA  #$ØØ
4B5D: 8D 52 48   857            STA  CRAUX+1     ;auf Ø
4B6Ø: A2 Ø3      858            LDX  #3
4B62: BD 9Ø BF   859  CREATE6   LDA  MLIDAT,X    ;Datum +
4B65: 9D 54 48   86Ø            STA  CRCDAT,X    ;Uhrzeit
4B68: CA         861            DEX
4B69: 1Ø F7      862            BPL  CREATE6
4B6B: AD 37 4B   863            LDA  CRSTORØ     ;DIR!
4B6E: 8D 53 48   864            STA  CRSTOR      ;Storage
4B71: 2Ø 45 48   865            JSR  CR
4B74: 9Ø D7      866            BCC  CREATE4
4B76: 8D CC 46   867            STA  ERRMLI
4B79: BØ D2      868            BCS  CREATE4
                 869  *--------------------------------
                 87Ø  *
                 871  * 5 = DELETE mit Parameter:
                 872  * ==========================
                 873  *
                 874  * Dateiname in NAME1 ablegen
                 875  *
4B7B: AD 43 46   876  DELETE    LDA  NAME1       ;leer?
4B7E: DØ ØD      877            BNE  DELETE3
4B8Ø: A9 82      878            LDA  #$82
4B82: 8D CD 46   879            STA  ERRALL
4B85: 4C 2Ø 47   88Ø  DELETE1   JMP  EXIT
4B88: 8D CC 46   881  DELETE2   STA  ERRMLI
4B8B: BØ F8      882            BCS  DELETE1
4B8D: 2Ø 58 48   883  DELETE3   JSR  DE
4B9Ø: 9Ø F3      884            BCC  DELETE1
4B92: BØ F4      885            BCS  DELETE2
                 886  *--------------------------------
                 887  *
                 888  * 6 = LOCK mit Parameter +
                 889  * 7 = UNLOCK mit Parameter:
                 89Ø  * ==========================
                 891  *
                 892  * Dateiname in NAME1 ablegen.
                 893  *
4B94: AD 43 46   894  LOCKUNL   LDA  NAME1       ;leer?
4B97: DØ ØD      895            BNE  LOCKUNL3
4B99: A9 82      896            LDA  #$82
4B9B: 8D CD 46   897            STA  ERRALL
4B9E: DØ Ø3      898            BNE  LOCKUNL2
4BAØ: 8D CC 46   899  LOCKUNL1  STA  ERRMLI
4BA3: 4C 2Ø 47   9ØØ  LOCKUNL2  JMP  EXIT
4BA6: 2Ø C8 4B   9Ø1  LOCKUNL3  JSR  GETSET
4BA9: BØ F5      9Ø2            BCS  LOCKUNL1
4BAB: AD CB 46   9Ø3            LDA  BEFEHL
4BAE: C9 Ø7      9Ø4            CMP  #7
4BBØ: FØ ØF      9Ø5            BEQ  LOCKUNL5
4BB2: AD 78 48   9Ø6            LDA  SFACC       ;6
4BB5: 29 21      9Ø7            AND  #%ØØ1ØØØØ1  ;Lock
4BB7: 8D 78 48   9Ø8  LOCKUNL4  STA  SFACC
4BBA: 2Ø 6E 48   9Ø9            JSR  SF
4BBD: BØ E1      91Ø            BCS  LOCKUNL1
4BBF: 9Ø E2      911            BCC  LOCKUNL2
4BC1: AD 78 48   912  LOCKUNL5  LDA  SFACC       ;7
4BC4: Ø9 C2      913            ORA  #%11ØØØØ1Ø  ;Unlock
4BC6: DØ EF      914            BNE  LOCKUNL4    ;stets
                 915  *
                 916  * Getset-Prefix für Lock/Unlock
                 917  *
4BC8: 2Ø 83 48   918  GETSET    JSR  GF
4BCB: BØ 23      919            BCS  GETSET2
4BCD: AD 8D 48   92Ø            LDA  GFACC
4BDØ: 8D 78 48   921            STA  SFACC       ;Access
4BD3: AD 8E 48   922            LDA  GFTYP
4BD6: 8D 79 48   923            STA  SFTYP       ;Type
4BD9: AD 8F 48   924            LDA  GFAUX
4BDC: 8D 7A 48   925            STA  SFAUX
4BDF: AD 9Ø 48   926            LDA  GFAUX+1
4BE2: 8D 7B 48   927            STA  SFAUX+1     ;Auxtype
4BE5: A2 Ø3      928            LDX  #3
4BE7: BD 94 48   929  GETSET1   LDA  GFMDAT,X
4BEA: 9D 7F 48   93Ø            STA  SFMDAT,X
4BED: CA         931            DEX
4BEE: 1Ø F7      932            BPL  GETSET1     ;Dat/Uhr
4BFØ: 6Ø         933  GETSET2   RTS
                 934  *--------------------------------
                 935  *
                 936  * 8 = RENAME mit Parameter:
                 937  * ==========================
                 938  *
                 939  * Altname in NAME1 und
                 94Ø  * Neuname in NAME2 eintragen.
                 941  *
4BF1: AD 43 46   942  RENAME    LDA  NAME1       ;leer?
```

```
4BF4: DØ Ø8      943            BNE  RENAME3
4BF6: A9 82      944            LDA  #$82
4BF8: 8D CD 46   945  RENAME1   STA  ERRALL
4BFB: 4C 2Ø 47   946  RENAME2   JMP  EXIT
4BFE: AD 84 46   947  RENAME3   LDA  NAME2
4CØ1: DØ Ø4      948            BNE  RENAME4
4CØ3: A9 83      949            LDA  #$83
4CØ5: DØ F1      95Ø            BNE  RENAME1
4CØ7: 2Ø 62 48   951  RENAME4   JSR  RN
4CØA: 9Ø EF      952            BCC  RENAME2
4CØC: 8D CC 46   953            STA  ERRMLI
4CØF: BØ EA      954            BCS  RENAME2
                 955  *--------------------------------
                 956  *
                 957  * 9 = BSAVE mit Parameter:
                 958  * ==========================
                 959  *
                 96Ø  * Dateiname in NAME1 eintragen.
                 961  * Anfangsadresse und Länge des
                 962  * Speicherbereichs in
                 963  * BANF und BLEN eintragen.
                 964  * Es wird BIN-Datei erzeugt.
                 965  * (ggf. unten ändern)
                 966  *
4C11: Ø6         967  BSAVETYP HEX  Ø6            ;BIN!
                 968  *
4C12: AD 43 46   969  BSAVE     LDA  NAME1       ;leer?
4C15: DØ ØE      97Ø            BNE  BSAVE4
4C17: A9 82      971            LDA  #$82
4C19: 8D CD 46   972  BSAVE1    STA  ERRALL
4C1C: 4C 2Ø 47   973  BSAVE2    JMP  EXIT
4C1F: 8D CC 46   974  BSAVE3    STA  ERRMLI
4C22: 4C 1C 4C   975            JMP  BSAVE2
4C25: 18         976  BSAVE4    CLC              ;Länge
4C26: AD C7 46   977            LDA  BLEN        ;null?
4C29: 6D C8 46   978            ADC  BLEN+1
4C2C: DØ Ø4      979            BNE  BSAVE5
4C2E: A9 84      98Ø            LDA  #$84
4C3Ø: DØ E7      981            BNE  BSAVE1
                 982  *
                 983  * Falls gleichnamige BIN-Datei
                 984  * existiert, dann erst löschen
                 985  *
4C32: 2Ø 83 48   986  BSAVE5    JSR  GF          ;Getinfo
4C35: BØ 11      987            BCS  BSAVE7      ;fehlt?
4C37: AD 8E 48   988            LDA  GFTYP
4C3A: CD 11 4C   989            CMP  BSAVETYP    ;BIN?
4C3D: FØ Ø4      99Ø            BEQ  BSAVE6
4C3F: A9 4A      991            LDA  #$4A        ;illegal
4C41: DØ DC      992            BNE  BSAVE3      ;Type!
4C43: 2Ø 58 48   993  BSAVE6    JSR  DE          ;Destroy
4C46: BØ D7      994            BCS  BSAVE3
                 995  *
                 996  * Jetzt neue Datei kreaten
                 997  *
4C48: AD 11 4C   998  BSAVE7    LDA  BSAVETYP
4C4B: 8D 5Ø 48   999            STA  CRTYP
4C4E: A9 ØØ      1ØØØ           LDA  #$ØØ        ;Auxtyp
4C5Ø: 8D 51 48   1ØØ1           STA  CRAUX       ;auf Ø
4C53: A9 ØØ      1ØØ2           LDA  #$ØØ
4C55: 8D 52 48   1ØØ3           STA  CRAUX+1
4C58: A9 Ø1      1ØØ4           LDA  #$Ø1        ;Sämling
4C5A: 8D 53 48   1ØØ5           STA  CRSTOR      ;Storage
4C5D: A2 Ø3      1ØØ6           LDX  #3
4C5F: BD 9Ø BF   1ØØ7 BSAVE8    LDA  MLIDAT,X    ;Datum +
4C62: 9D 54 48   1ØØ8           STA  CRCDAT,X    ;Uhrzeit
4C65: CA         1ØØ9           DEX
4C66: 1Ø F7      1Ø1Ø           BPL  BSAVE8
4C68: 2Ø 45 48   1Ø11           JSR  CR          ;Create
4C6B: BØ B2      1Ø12           BCS  BSAVE3
                 1Ø13 *
                 1Ø14 * Open, Newline und Write
                 1Ø15 *
4C6D: 2Ø BB 48   1Ø16           JSR  OP
4C7Ø: BØ 2B      1Ø17           BCS  BSAVE9
4C72: AD C7 48   1Ø18           LDA  OPREF       ;File-R.
4C75: 8D DØ 48   1Ø19           STA  NLREF       ;übertr.
4C78: 8D EA 48   1Ø2Ø           STA  WRREF
4C7B: 2Ø C8 48   1Ø21           JSR  NL
4C7E: BØ 1D      1Ø22           BCS  BSAVE9
4C8Ø: AD C5 46   1Ø23           LDA  BANF
4C83: 8D EB 48   1Ø24           STA  WRDPUF
4C86: AD C6 46   1Ø25           LDA  BANF+1
4C89: 8D EC 48   1Ø26           STA  WRDPUF+1
4C8C: AD C7 46   1Ø27           LDA  BLEN
4C8F: 8D ED 48   1Ø28           STA  WRREQ
4C92: AD C8 46   1Ø29           LDA  BLEN+1
```

```
4C95: 8D EE 48  1030          STA  WRREQ+1
4C98: 20 E2 48  1031          JSR  WR
4C9B: 90 03     1032          BCC  BSAVE10
4C9D: 8D CC 46  1033 BSAVE9   STA  ERRMLI
                1034 *
                1035 * Close wird immer versucht
                1036 *
4CA0: 20 F1 48  1037 BSAVE10  JSR  CL
4CA3: 4C 1C 4C  1038          JMP  BSAVE2
                1039 *--------------------------------
                1040 *
                1041 * 10 = BLOAD mit Parameter:
                1042 * ==========================
                1043 *
                1044 * Dateiname in NAME1 eintragen.
                1045 * Anfangsadresse und max. Länge
                1046 * des Speicherbereichs in
                1047 * BANF und BLEN eintragen.
                1048 * Es wird BIN-Datei erwartet.
                1049 * (ggf. unten ändern)
                1050 *
                1051 * Nachher enthält FLEN die
                1052 * effektiv eingelesene Länge,
                1053 * die kleiner/gleich BLEN ist.
                1054 * Wenn < BLEN, dann ERRALL = $85
                1055 *
4CA6: 06        1056 BLOADTYP HEX  06          ;BIN!
                1057 *
4CA7: AD 43 46  1058 BLOAD    LDA  NAME1       ;leer?
4CAA: D0 0E     1059          BNE  BLOAD4
4CAC: A9 82     1060          LDA  #$82
4CAE: 8D CD 46  1061 BLOAD1   STA  ERRALL
4CB1: 4C 20 47  1062 BLOAD2   JMP  EXIT
4CB4: 8D CC 46  1063 BLOAD3   STA  ERRMLI
4CB7: 4C B1 4C  1064          JMP  BLOAD2
4CBA: 18        1065 BLOAD4   CLC              ;Länge
4CBB: AD C7 46  1066          LDA  BLEN        ;null?
4CBE: 6D C8 46  1067          ADC  BLEN+1
4CC1: D0 04     1068          BNE  BLOAD5
4CC3: A9 84     1069          LDA  #$84
4CC5: D0 E7     1070          BNE  BLOAD1
                1071 *
4CC7: 20 83 48  1072 BLOAD5   JSR  GF          ;Getinfo
4CCA: B0 E8     1073          BCS  BLOAD3
4CCC: AD 8E 48  1074          LDA  GFTYP
4CCF: CD A6 4C  1075          CMP  BLOADTYP    ;BIN?
4CD2: F0 04     1076          BEQ  BLOAD6
4CD4: A9 4A     1077          LDA  #$4A        ;illegal
4CD6: D0 DC     1078          BNE  BLOAD3      ;Type!
                1079 *
                1080 * Open und Newline
                1081 *
4CD8: A9 00     1082 BLOAD6   LDA  #0          ;löschen
4CDA: 8D E0 48  1083          STA  RDTRA       ;wegen
4CDD: 8D E1 48  1084          STA  RDTRA+1     ;früher
4CE0: 20 BB 48  1085          JSR  OP
4CE3: B0 5F     1086          BCS  BLOAD10
4CE5: AD C7 48  1087          LDA  OPREF       ;File-R.
4CE8: 8D D0 48  1088          STA  NLREF       ;übertr.
4CEB: 8D 2F 49  1089          STA  GEREF
4CEE: 8D DB 48  1090          STA  RDREF
4CF1: 20 C8 48  1091          JSR  NL
4CF4: B0 4E     1092          BCS  BLOAD10
                1093 *
                1094 * Bei EOF > $FFFF nichts lesen
                1095 *
4CF6: 20 27 49  1096          JSR  GE          ;LLMMHH
4CF9: B0 49     1097          BCS  BLOAD10
4CFB: AD 32 49  1098          LDA  GEEOF+2
4CFE: F0 07     1099          BEQ  BLOAD7      ;<$FFFF
4D00: A9 85     1100          LDA  #$85
4D02: 8D CD 46  1101          STA  ERRALL
4D05: D0 40     1102          BNE  BLOAD11
                1103 *
4D07: AD 30 49  1104 BLOAD7   LDA  GEEOF       ;RDREQ=
4D0A: 8D DE 48  1105          STA  RDREQ       ;EOF
4D0D: AD 31 49  1106          LDA  GEEOF+1
4D10: 8D DF 48  1107          STA  RDREQ+1
4D13: 38        1108          SEC              ;Abzieh-
4D14: AD C7 46  1109          LDA  BLEN        ;test
4D17: ED 30 49  1110          SBC  GEEOF
4D1A: AD C8 46  1111          LDA  BLEN+1
4D1D: ED 31 49  1112          SBC  GEEOF+1
4D20: B0 11     1113          BCS  BLOAD9
                1114 *
4D22: AD C7 46  1115 BLOAD8   LDA  BLEN        ;RDREQ=
4D25: 8D DE 48  1116          STA  RDREQ       ;BLEN
4D28: AD C8 46  1117          LDA  BLEN+1
4D2B: 8D DF 48  1118          STA  RDREQ+1
4D2E: A9 85     1119          LDA  #$85
4D30: 8D CD 46  1120          STA  ERRALL
                1121 *
                1122 * Read Soll- oder Ist-Länge
                1123 *
4D33: AD C5 46  1124 BLOAD9   LDA  BANF
4D36: 8D DC 48  1125          STA  RDDPUF
4D39: AD C6 46  1126          LDA  BANF+1
4D3C: 8D DD 48  1127          STA  RDDPUF+1
4D3F: 20 D3 48  1128          JSR  RD
4D42: 90 03     1129          BCC  BLOAD11
4D44: 8D CC 46  1130 BLOAD10  STA  ERRMLI
                1131 *
                1132 * Close immer versuchen
                1133 *
4D47: 20 F1 48  1134 BLOAD11  JSR  CL
4D4A: AD E0 48  1135          LDA  RDTRA
4D4D: 8D C9 46  1136          STA  FLEN
4D50: AD E1 48  1137          LDA  RDTRA+1
4D53: 8D CA 46  1138          STA  FLEN+1
4D56: 4C B1 4C  1139          JMP  BLOAD2
Länge 3417 Bytes
```

PRODOS.LIB.DEMO

für Applesoft unter dem BASIC.SYSTEM 1.1, das jedoch nach dem Programmstart nicht mehr benötigt wird.

```
10 HOME : INVERSE : PRINT "PRODOS.LIB.DEMO": NORMAL : PRINT
   CHR$ (4)"BLOAD PRODOS.LIB": POKE 242,0: PR# 0: IN# 0:
   GOSUB 70: POKE N0,0: POKE B,2: CALL E: PRINT :
   IF  PEEK (N0) = 0 THEN  PRINT "PREFIX FEHLT": END
14 CLEAR : DATA
   " 0=REBOOT "," 1=ONLINE "," 2=PREFIX ",
   " 3=CATALOG"," 4=CREATE "," 5=DELETE ",
   " 6=LOCK   "," 7=UNLOCK "," 8=RENAME ",
   " 9=BSAVE  ","10=BLOAD  ","11=ENDE"
18 FOR X = 0 TO 11: READ X$: PRINT X$;: NEXT : PRINT : PRINT:
   GOSUB 70
22 INPUT "BEFEHLSNR.:";X$:C =  VAL (X$):
   ON C = 0 AND X$ < > "0" GOTO 14: ON C < 0 OR C > 11
   GOTO 14: PRINT : IF C = 11 THEN  END
26 POKE B,C: REM BEFEHL
30 IF C = 0 OR C = 1 THEN  CALL E: GOTO 50: REM REBOOT,ONLINE
34 IF C = 2 THEN X$ = "PREFIX: ":N = N0: GOSUB 62: CALL E:
   GOTO 50: REM PREFIX
38 X$ = "NAME: ":N = N1: IF C > 2 AND C < 8 THEN  GOSUB 62:
    CALL E: GOTO 50: REM CAT,CREATE,DELETE,LOCK,UNLOCK
42 IF C = 8 THEN X$ = "ALTNAME: ":N = N1: GOSUB 62:
   X$ = "NEUNAME: ":N = N2: GOSUB 62: CALL E: GOTO 50:
   REM RENAME
46 IF C = 9 OR C = 10 THEN X$ = "NAME: ":N = N1: GOSUB 62:
   POKE BA,0: POKE BA + 1,32: POKE BL,0: POKE BL + 1,32:
   CALL E: REM A$2000,L$2000
50 PRINT : IF C = 10 THEN  PRINT 8192;"-"; PEEK (FL) +
   PEEK (FL + 1) * 256: REM $2000?
54 X =  PEEK (EA): IF X < > 0 THEN PRINT "ERR-ALL: 8";X - 128
58 PRINT : GOTO 14
62 PRINT X$;: INPUT "";N$:L =  LEN (N$): ON L > 64 GOTO 62:
   POKE N,L: IF L = 0 THEN  RETURN
66 FOR X = 1 TO L: POKE N + X, ASC ( MID$ (N$,X,1)): NEXT:
   RETURN
70 N0 = 17920:N1 = 17987:N2 = 18052:BA = 18117:BL = 18119:
   FL = 18121:B = 18123:EA = 18125:E = 18126: RETURN
```


Telefonische Bestellungen?

Da unsere Peeker-Disketten in offener Rechnung und nicht in dem für Sie teuren Nachnahme-Verfahren ausgeliefert werden, haben Sie bitte Verständnis dafür, daß wir **nur noch schriftliche Bestellungen annehmen.**

Sie können dazu beispielsweise die in jedem Peeker eingehefteten Bestellkarten verwenden.

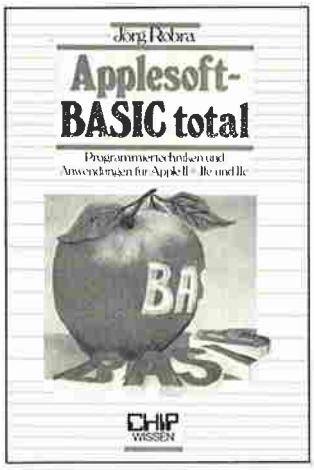
Hüthig Software Service

PRODOS.LIB.P

für Kyan-Pascal 2.Ø unter PRODOS 1.1.1

```
#A
_USESHIRES
#

PROGRAM PRODOSLIB;

TYPE FILENAME = ARRAY [1..64] OF CHAR;

VAR   PREFIX,  NAME1,  NAME2:  FILENAME;
      BANF,    BLEN,   FLEN:   INTEGER;
      BEFEHL,  ERRMLI, ERRALL: CHAR;

      I: INTEGER;

{-----------------------------------------------}

PROCEDURE PRODOS
 (VAR PREFIX,  NAME1,  NAME2:  FILENAME;
  VAR BANF,    BLEN,   FLEN:   INTEGER;
  VAR BEFEHL,  ERRMLI, ERRALL: CHAR);
BEGIN
#A
; Assembler-Routinen PRODOS.LIB nicht gelistet!
#
END;

{-----------------------------------------------}

BEGIN

{Werte initialisieren und Online ausfuehren}

WRITELN (CHR(12)); {Home}
WRITELN ('ProDOS-Library fuer Kyan 2.Ø');
WRITELN ('von Ulrich Stiehl  1Ø.5.1986');

FOR I := 1 TO 64 DO
 BEGIN
 PREFIX[I] := ' '; NAME1[I] := ' '; NAME2[I] := ' '
 END;
BANF := 8192;   BLEN := 8192;     FLEN := Ø;
BEFEHL := 'B';  ERRMLI := CHR(Ø); ERRALL := CHR(Ø);

PRODOS (PREFIX, NAME1,  NAME2,
        BANF,   BLEN,   FLEN,
        BEFEHL, ERRMLI, ERRALL);

WRITELN; WRITELN; WRITELN ('WARNUNG:');
WRITELN ('Erst Prefix setzen!');
WRITELN ('Jedoch kein Null-Prefix!');

REPEAT
 WRITELN;
 WRITE ('A Reboot  ');
 WRITE ('B Online  ');
 WRITE ('C Prefix  ');
 WRITE ('D Catalog ');
 WRITE ('E Create  ');
 WRITE ('F Delete  '); WRITELN;
 WRITE ('G Lock    ');
 WRITE ('H Unlock  ');
 WRITE ('I Rename  ');
 WRITE ('J Bsave   ');
 WRITE ('K Bload   ');
 WRITE ('L Ende    '); WRITELN;
 WRITE ('Befehl (A-L): '); READLN (BEFEHL);

 IF (BEFEHL = 'C') THEN
  BEGIN WRITE ('PREFIX: '); READLN (PREFIX) END;
 IF (BEFEHL > 'C') AND (BEFEHL < 'L') THEN
  BEGIN WRITE ('NAME: '); READLN (NAME1) END;
 IF (BEFEHL = 'I') THEN
  BEGIN WRITE ('NEUNAME: '); READLN (NAME2) END;

 PRODOS (PREFIX, NAME1,  NAME2,
         BANF,   BLEN,   FLEN,
         BEFEHL, ERRMLI, ERRALL);

 IF (ERRALL <> CHR(Ø)) THEN
  WRITELN ('ERRALL: 8', ORD (ERRALL) - 128);
 IF (BEFEHL = 'K') AND (ERRMLI = CHR(Ø)) THEN
  WRITELN ('8192-', FLEN);

UNTIL BEFEHL = 'L';
END.
```


VOGEL Computer-bücher



Robra, Jörg
Applesoft-BASIC total
Programmiertechniken und Anwendungen für Apple II+, IIe und IIc
340 Seiten, zahlr. Abb., 45 DM
ISBN 3-8023-**0872**-8

Das Buch zeigt, wie aus einem Problem — zunächst computerunabhängig — über Analyse, Zieldefinition, Benutzerhandbuch, Modularisierung und strukturierte Programmierung ein sauber aufgebautes und benutzerfreundliches Programm entsteht. Nach einem Überblick zum Aufbau von DOS 3.3 werden aufwendige Programme verschiedener Sachgebiete einschließlich Grafik schrittweise entwickelt.

Kretschmer, Bernd
Multiplan-Training auf Apple IIe und Apple IIc
Eine leicht lesbare systematische Einführung
260 Seiten, zahlr. Abb., 38 DM
ISBN 3-8023-**0847**-6

Alle Nutzer des Apple II werden hier ein Arbeitsmittel finden, das alles Wesentliche über das Tabellenkalkulationsprogramm Multiplan aussagt. Diese anschauliche Einführung ist genau auf diese Zielgruppe abgestimmt und gibt schon im didaktischen Ansatz alle Möglichkeiten, Schritt für Schritt anhand nachvollziehbarer Beispiele den Lernstoff zu verstehen und zu verarbeiten.

Bücher zum Apple

Senftleben, Dietrich
Start mit Apple-Logo für Apple II, IIe und IIc
Das kleine Logo-Einmaleins
Grafik · Text · Musik
222 Seiten, 70 Abb., 35 DM
ISBN 3-8023-**0832**-8

Willkommen bei Logo und seinen beiden Versionen Apple-Logo und Apple-Logo II, bekannt als benutzerfreundliche Computersprache für Ausbildung, Freizeit und Beruf. Mittels Schildkrötengrafik wird das kleine Logo-Einmaleins in 12 Lektionen beschrieben. Das Buch verlangt praktische Mitarbeit. Es hat seinen Platz neben dem Computer und gibt Hilfen und Anregungen für eigenes Forschen.

Renner, Gerhard
Turbo Pascal
Versionen 1.xx bis 3.xx
Einführung, Sprachdefinition und Programmierung für Ausbildung, Hobby und Beruf
308 Seiten, 131 Abb., 43 DM
ISBN 3-8023-**0758**-5

Turbo Pascal ist im Vergleich zu anderen Pascal-Dialekten besonders benutzerfreundlich und leistungsfähig, dazu als preiswerter „Volkscompiler" interessant für jedermann. Die Programmentwicklung und -strukturierung, die Sprachbeschreibung mit allen Standardvereinbarungen (auch Grafik, Turtlegrafik, Windows) werden Studierenden und Hobbyprogrammierern anschaulich vermittelt.
Auch auf Diskette erhältlich!

Sie erhalten bei Ihrem Buch- und Computerfachhändler kostenlos das neue Verzeichnis „VOGEL-Computerbücher 1986" mit ca. 100 Titeln.

VOGEL-BUCHVERLAG
WÜRZBURG

VOGEL-Computerbücher — mehr wissen, mehr leisten.

Datenübertragungsraten beim Macintosh

von Arne Schirmacher

Der Macintosh ist als langsamer Computer in Verruf geraten. Da stellt sich die Frage, ob es an unpassenden Versuchsbedingungen oder am Computer selbst liegt. In ihm ist ja bekanntlich der 68 000-Prozessor eingebaut, der mit etwa 5.5 MHz für RAM-Programme und 8 MHz für ROM-Routinen läuft. Das sollte eigentlich für eine ziemlich hohe Rechengeschwindigkeit ausreichen.
Thema dieses Artikels ist die Input-Output-Geschwindigkeit des Betriebssystems. Ich habe dazu ein kleines Assemblerprogramm geschrieben, das zehnmal einen Datenblock von 100K Länge auf verschiedene blockorientierte Massenspeicher schreiben kann. Welche Geschwindigkeiten dabei herauskommen, können Sie der Tabelle entnehmen. Alle gemessenen Werte beziehen sich auf einen Macintosh mit 512K RAM ohne weitere Hardware, mit den alten 64K ROMs und dem alten MFS (Macintosh-File-System, in etwa gleich dem DOS anderer Systeme).
Die Laufzeit des Programms beträgt 38 s, wenn es das interne oder externe normale 400K-Laufwerk liest. Daraus ergibt sich eine Datenübertragungsrate von etwa 26K/s. Keine atemberaubende Geschwindigkeit; aber man sollte berücksichtigen, daß die Datenübertragung zu den normalen Laufwerken über eine serielle Schnittstelle geschieht. Wer nachrechnet, kommt auf ungefähr 250 KBaud Übertragungsgeschwindigkeit, und das ist für eine einfache serielle Schnittstelle schon recht anständig. Manche Festplatten benutzen diese serielle Schnittstelle mit einer noch höheren Baudrate. Der neue Macintosh Plus hat eine parallele Schnittstelle, den sog. SCSI-Port. Man kann davon ausgehen, daß diese Schnittstelle um einiges schneller arbeitet als die serielle Schnittstelle; die Übertragungsgeschwindigkeit wird sich bei den normalen 400K-Laufwerken also erhöhen. Die neuen 800K-Laufwerke haben möglicherweise auch eine höhere Arbeitsgeschwindigkeit.
Unschlagbar ist natürlich eine RAM-Disk. Das Testprogramm liest die insgesamt 1 M Daten derartig schnell von der RAM-Disk, daß ich 100 Durchläufe mit je 100K programmieren mußte, um überhaupt etwas messen zu können. Die Datenübertragungsrate von 1.1 M/s liegt dicht bei dem Wert, den der 68 000-Prozessor bei 5.5 MHz überhaupt schaffen kann. Es gibt übrigens zusätzliche RAM-Boards, die bis zu 2 M Speicher haben und eine Taktfrequenz von 8 bis zu 12.5 MHz erlauben.

Organisation des I/O-Systems

Nun zum Programm selbst. Ich verwende für meine Assemblerprogramme ausschließlich den MacASM-Assembler, der erheblich schneller arbeitet als Apples MDS-Assembler. Außerdem ist es möglich, direkt ins RAM zu assemblieren, was für das Programm Speedtest sehr praktisch ist. Glücklicherweise gibt es für alles und jedes im Macintosh ROM-Routinen, natürlich auch für das I/O-System. Normalerweise kommuniziert ein Programm mit der Peripherie über den sogenannten File-Manager, der ein recht komfortables DOS darstellt. Der File-Manager greift über den Device-Manager auf die einzelnen Peripheriegeräte zu. Und der Device-Manager verwendet seinerseits hardwareabhängige Programme. Für die gewöhnlichen 3.5"-Disk-Drives von Apple ist diese hardwareabhängige Software der sog. Disk-Driver. Er ist, genau wie der File-Manager und der Device-Manager, im ROM vorhanden, weil ja ohnehin mindestens ein Diskettenlaufwerk vorhanden ist. Wenn andere Peripheriegeräte angeschlossen werden, dann wird mit einem Installationsprogramm ein eigener Driver ins RAM geladen. Das ist z. B. bei den RAM-Disks und bei Hard-Disks der Fall. Diese Driver werden genauso angesprochen wie der Disk-Driver, daher muß ein Programm sich nicht selbst an die unterschiedliche Hardware anpassen.
Der Device-Driver kommuniziert mit dem Programm des Anwenders über den sog. Parameterblock. In ihm werden alle für den jeweiligen Aufruf benötigten Parameter eingetragen. Nach Rückkehr von der ROM-Routine sind dann die Ergebnisse des Aufrufs ebenfalls in diesem Block eingetragen. Vor Aufruf der Routine wird das Register A0 des Prozessors mit einem Pointer auf den Parameterblock geladen, dann wird die Routine (hier entweder Read oder Write) aufgerufen. Das geschieht nicht durch ein JSR, sondern durch einen sogenannten $A-Trap. Ein $A-Trap ist eine Art Software-Interrupt, der in eine Sprungtabelle verzweigt. Hier in dieser Sprungtabelle ist dann die Adresse der ROM-Routine enthalten. Wenn das Programm aus dieser Routine zurückkehrt, enthält das Register D0 einen 16-Bit-Wert, der entweder 0 oder einen Fehlercode enthält. Alle anderen Register des Prozessors sind unverändert.

Diskettensektoren lesen und schreiben

Die erste Anweisung des Programms bereitet den Schleifenzähler vor. Das Register D1 enthält die Anzahl der Durchläufe −1; hier enthält es den Wert 9 für 10 Durchläufe. Dann wird der Parameterblock vorbereitet. Es gibt einen ausreichend langen Platz im unteren RAM, den auch das Betriebssystem verwendet, natürlich könnte man den benötigten Platz auch im Stack oder sonstwo belegen. Die Anweisung in Zeile 4100 lädt einen Pointer auf diesen Speicherbereich in das Register A0.
In den folgenden Zeilen wird der Parameterblock ausgefüllt: Die Laufwerksnummer ist 1 für das interne, 2 für das externe Laufwerk, 3 oder eine andere Nummer für RAM-Disks, Harddisks usw. Die Driver-Reference-Nummer ist −5 für den normalen Sony-Driver oder −11 für die von mir verwendete RAM-Disk. Dann wird die Adresse des Puffers eingetragen, in den die gelesenen Bytes abgelegt werden. Aber Vorsicht: Man muß selbst dafür sorgen, daß der benötigte Platz tatsächlich vorhanden ist. Auf einem Macintosh mit 128K kann man nicht 100K belegen. Wenn man einen 512K-Macintosh verwendet, kann man nicht 340K davon für die RAM-Disk verwenden. In diesem Beispiel beginnt der Puffer unmittelbar hinter dem Assemblerprogramm; MacASM hat genügend Speicherplatz bereitgestellt.
Dann folgen Informationen, wieviele Bytes gelesen werden sollen (100 K) und an welcher Stelle auf der Diskette sie sich befinden. Eine 1 im Feld IoPosMode bestimmt, daß relativ zum Diskettenbeginn gelesen werden soll, und IoPosOffset enthält den Offset zum Diskettenbeginn. Hier ist er 0, es wird also beim ersten Byte der Diskette beginnend gelesen.
Dann folgt der Aufruf der Device-Manager-Routine Read. Wie gesagt, hier geschieht es mit Hilfe eines $A-Traps. Wer die Daten schreiben will (Achtung: Eine Leerdiskette ins Laufwerk legen), schreibt dort einfach das in Zeile 2700 definierte Write hin.

Nach Aufruf der Routine liest bzw. schreibt der Driver sein Peripheriegerät. Wenn kein Fehler auftritt, enthält Register D0 den Wert 0 (16 Bits), sonst eine negative Fehlernummer. Eventuell könnte man noch eine kleine Routine anhängen, die den Programmierer vom Auftreten eines Fehlers informiert. Man kann aber auch ein paar Durchläufe mit einem Debugger in Einzelschritten durchführen und selbst sehen, welchen Wert das Register D0 enthält. Wenn es beim ersten Mal den Wert 0 enthält, sollte es eigentlich auch in allen anderen Durchgängen denselben Wert haben.

Weitere Peripheriegeräte

Es wäre interessant, auch Festplatten oder den Macintosh Plus mit diesem Programm zu testen. Da ich selbst nicht die Hardware dazu habe, hier Informationen, wie man das selbst machen kann.
Erste Voraussetzung ist natürlich, daß man den MacASM-Assembler hat. Dann kann man das Programm eintippen und assemblieren lassen, aber man muß für weitere Peripheriegeräte ihre Reference-Nummer und die Laufwerksnummer kennen. Um diese Werte herauszubekommen, muß man ein wenig im Betriebssystem herumstöbern. Das geht am besten mit einem Debugger, z. B. Apples MacsBug. Wer nicht an dieses nützliche Programm herankommt, kann sich die Bytes auch von BASIC aus mit Peek ansehen. Er muß sich dann ein kleines Programm dazu schreiben.
Man kopiert die Datei MacsBug auf seine Startdiskette. Manchmal heißt sie auch xMacsBug oder Maxbug. In diesem Fall muß man sie in MacsBug umbenennen. Dann wird der Macintosh mit dieser Startdiskette neu gebootet. Der Debugger wird automatisch geladen. Nun lädt man die für die RAM-Disk oder Harddisk usw. erforderliche Software und drückt auf den Interruptknopf (der hintere von beiden). Jetzt läuft der Debugger. Nun tippt man ein:
DM 308 <RETURN>
Man sieht eine Reihe von Ziffern und Buchstaben in Gruppen: Das sind Adressen und Speicherinhalte. An der Adresse $ 308 befindet sich der Drive-Queue-Header. Die zweite und dritte Vierergruppe stellt die 32-Bit-Adresse eines Eintrags für ein Laufwerk dar. Das wird in der Regel das Laufwerk Nr. 1 (internes Laufwerk) sein. Jetzt gibt man DM und eben diese Adresse ein. Das ist der Eintrag für dieses Laufwerk. Die ersten beiden Vierergruppen sind die Adresse des nächsten Eintrags (oder Null beim letzten Eintrag), die vierte Vierergruppe ist die Laufwerksnummer, die fünfte die Reference-Nummer. Das schreibt man sich auf und listet sich die weiteren Einträge in der Drive-Queue. Diese Informationen kann man nun in das Assemblerprogramm einbauen und jedes beliebige Peripheriegerät testen.
Weil ich gerade dabei bin, noch in Stichworten ein paar weitere MacsBug-Befehle: SM schreibt Bytes an eine Adresse, IL disassembliert 68 000-Code, T führt einen oder mehrere Einzelschritte durch, BR und CL setzen bzw. löschen einen Breakpoint an der angegebenen Adresse, G führt das Programm an der Stelle fort, an der es unterbrochen wurde. Viel Spaß!

Speedtest

```
00100 ;save Speedtest.Asm
00200 ***********************************************************************
00300 *                                                                     *
00400 * Dieses Programm bestimmt die I/O-Geschwindigkeit des Apple         *
00500 * Macintosh Computers für verschiedene Peripheriegeräte und           *
00600 * Betriebssysteme. Es liest bzw. schreibt zehnmal einen Datenblock   *
00700 * von 100 KByte Länge auf ein blockorientiertes Peripheriegerät.      *
00800 * Das Peripheriegerät wird durch die Laufwerksnummer und seine       *
00900 * refNum identifiziert: Normale Macintosh Drive's (400 KByte)         *
01000 * haben Laufwerksnummern von 1-2 und eine refNum von -5. Die         *
01100 * verwendete RAMDisk hat eine Laufwerksnummer von 3 und eine         *
01200 * refNum von -11.                                                     *
01300 *                                                                     *
01400 * Gemessene Zeiten:                                                   *
01500 *                                                                     *
01600 * Lesen internes 400 KByte Laufwerk       38  s    26  KByte/s        *
01700 * Schreiben internes 400 KByte Laufwerk   65  s    15  KByte/s        *
01800 * Lesen mit Zusatzprogramm TurboCharger   1.5 s    666 KByte/s        *
01900 * Schreiben mit TurboCharger              13  s    77  KByte/s        *
02000 * Lesen auf RAMDisk                       0.9 s    1.1 MByte/s        *
02100 * Schreiben auf RAMDisk                   0.9 s    1.1 MByte/s        *
02200 *                                                                     *
02300 * Arne Schirmacher, 6070 Langen                                       *
02400 ***********************************************************************
02500 ;
02600 Read              EQU  $A002
02700 Write             EQU  $A003
02800 ;
02900 ioVRefNum         EQU  22
03000 ioRefNum          EQU  24
03100 ioBuffer          EQU  32
03200 ioReqCount        EQU  36
03300 ioPosMode         EQU  44
03400 ioPosOffset       EQU  46
03500 ;
03600 Params            EQU  $3A4
03700 ;
03800          org      $20000
03900 ;
04000 Start    MOVEQ    #9,D1                    Anzahl der Durchläufe
04100 Loop     LEA      Params,A0                Adresse Parameterblock
04200          LEA      Buffer,A1                Adr. Speicherblock für I/O
04300          MOVE.W   #2,ioVRefNum(A0)         Laufwerksnummer (1-3)
04400          MOVE.W   #-5,ioRefNum(A0)         -5: Disk, -11: RAMDisk
04500          MOVE.L   A1,ioBuffer(A0)          Quelle / Ziel der Daten
04600          MOVE.L   #100*1024,ioReqCount(A0) 100 KByte lesen / schreiben
04700          MOVE.W   #1,ioPosMode(A0)         bei Diskanfang beginnen
04800          CLR.L    ioPosOffset(A0)          0 Bytes vom Anfang entfernt
04900          DATA     /Read                    OS-Routine für Lesen
05000          DBRA     D1,Loop                  nächster Schleifendurchlauf
05100          RTS
05200 Buffer
```

Tabelle 1

Peripheriegerät, IM Lesen/ Schreiben	Zeit/s	K/s
Lesen internes 400-K-Laufwerk	38	26
Schreiben internes 400-K-Laufwerk	65	15
Lesen mit Zusatzprogramm TurboCharger	1.5	666
Schreiben mit TurboCharger	13	77
Lesen von RAM-Disk	0.9	1110
Schreiben auf RAM-Disk	0.9	1110

Schreiben mit der Maus

Textverarbeitungsprogramm Mousewrite

getestet von Thomas Bühner

Der Kauf eines Computerprogramms ist immer mit Risiken verbunden. Man steht vor ähnlichen Problemen wie bei der Anschaffung eines Autos: Die Probefahrt beim Händler ist zwar möglich; viele Schwächen werden sich aber erst nach monatelangem Betrieb herausstellen. Besonders offensichtlich wird dieses Dilemma bei dem US-amerikanischen Textverarbeitungsprogramm Mousewrite, dessen Besitzer sich meist so sehr über den Bedienungskomfort freuen wird, daß ihm die Unzulänglichkeiten zu spät auffallen.

Standpunkt

Damit der Leser dieses Berichts einen ungefähren Eindruck der Situation bekommt, aus der das Programm unter die Lupe genommen wurde, hier ein kurzer Abriß der Textanwendungen des Autors: Jede Woche werden ein bis zwei Briefe geschrieben, die einen Umfang von etwa einer halben bis zwei Schreibmaschinenseiten haben (Forderung: schnelle Abwicklung und gutes Layout); alle vierzehn Tage bringt der Verfasser einen Zeitschriftenartikel zu Papier (Forderung: genaue Gesamtlänge des Textes feststellen); außerdem fallen alle vier Wochen einseitige Serienbriefe an, die für zehn bis zwanzig verschiedene Adressaten bestimmt sind und je nach Empfänger einen etwas abgewandelten Inhalt haben (Forderung: leistungsfähige Serienbriefgestaltung und gutes Layout). All diese Texte und das bisher einzige Buch des Autors (Forderung: große Textlänge) werden mit einem Programm erledigt, das ca. DM 200 kostet.

Dilemma

Leidgeprüften Vielschreibern ist jetzt schon klar, daß die vom Autor verwendete Software *eine* gute Eigenschaft mit Sicherheit *nicht* hat: leichte Bedienbarkeit. Bisher mußte man sich immer entscheiden, ob es wichtiger war, möglichst effektiv oder vor allem bequem zu arbeiten. Hat sich diese Situation nun mit der Entwicklung von Mousewrite zugunsten des Anwenders verändert? Wir werden sehen.

1. Notwendige Ausstattung

Will man Mousewrite zur Textverarbeitung einsetzen, muß man entweder einen Apple IIc sein eigen nennen oder den IIe zweifach aufrüsten:

- Man braucht eine 80-Zeichen-Karte mit 64K zusätzlichem Speicherplatz.
- Der IIe muß mit dem neuen Zeichensatz-ROM bestückt werden.

Dieses ROM ist ein kleiner Chip, auf dem sich Informationen über die Form aller Buchstaben befinden, die auf dem Bildschirm dargestellt werden können. Unter den Zeichen des „neuen" ROMs sind auch grafische Symbole – wie Pfeile, Äpfel und Rahmenteile –, mit deren Hilfe man einen übersichtlicheren Bildschirmaufbau erhalten kann. Ob der eigene Apple IIe schon mit diesem ROM ausgerüstet ist, sieht man bereits kurz nach dem Start des Programms: Bestehen die Balken und Rahmen, die man zu Gesicht bekommt, aus schwarzen Buchstaben oder Satzzeichen auf weißem Hintergrund, muß man das ROM für etwa DM 200 beim örtlichen Apple-Händler austauschen lassen. Trotz des verhältnismäßig hohen Preises für die Aufrüstung kann das durchaus lohnend sein, da in Zukunft immer mehr Programme diese sogenannten „Mauszeichen" verwenden werden. Der Begriff „Mauszeichen" stammt daher, daß die grafischen Symbole nur bei Software verwendet werden, deren Bedienung durch eine Maus erfolgt.
Zwar kann Mousewrite auch ohne eine Maus – über die Tastatur – bedient werden, mitunter gestaltet sich die Arbeit dann

Apple IIc

Handbuch für Anwender und Programmierer

von Carl-Ulrich Wassermann

1985, 324 S., zahlr. Abb., kart., DM 35,—
ISBN 3-7785-1157-2

Wenn Sie die Leistungsfähigkeit Ihres Apple IIc bisher noch nicht ausschöpfen konnten, brauchen Sie dieses Buch.

aber sehr umständlich. Eine optische Maus ist der üblichen mechanischen vorzuziehen (siehe auch Testbericht in Peeker 4/86, Seite 67 f.).

Besitzt man eine Harddisk, die mit dem Betriebssystem ProDOS arbeitet, kann man alle für die Funktion des Programms notwendigen Dateien auf die Festplatte übertragen und so die Arbeit angenehmer gestalten.

Auch Dokumente, die eventuell mit einem zuvor verwendeten anderen Textverarbeitungsprogramm auf Diskette gespeichert wurden, können von Mousewrite bearbeitet werden, solange sie in der Form von Textdateien vorliegen und nicht als Binärfibles abgelegt wurden.

2. Handbuch

Das 230 Seiten umfassende englischsprachige Handbuch besteht aus drei Teilen: Eine hundertseitige Einführung zeigt dem unerfahrenen Benutzer mit Hilfe vieler Beispiele alle Mousewrite-Funktionen im praktischen Einsatz. Um das Lernen einfacher zu gestalten, arbeitet man mit schon fertig auf Diskette vorliegenden Übungstexten. Die Sprache ist klar und kommt weitgehend ohne technische Begriffe aus, die das Verständnis vieler Handbücher erschweren.

Für schon erfahrene Anwender von Textverarbeitungsprogrammen sind weitere siebzig Seiten vorgesehen, in denen alle Funktionen noch einmal knapp und übersichtlich beschrieben werden. Hat man als Anfänger die ersten hundert Seiten einmal durchgearbeitet, wird man in Zukunft meist den zweiten, kürzeren Teil zu Rate ziehen, wenn Unklarheiten auftauchen.

Hat schließlich jemand schon längere Zeit mit Mousewrite gearbeitet und wieder vergessen, wie seltener gebrauchte Dinge funktionieren, können auf weiteren dreißig Seiten alle Programmoptionen noch einmal nachgeschlagen werden. Die Sprache dieses Teils ist notwendigerweise sehr technisch; nach den ersten Wochen der Benutzung dürfte aber auch ein EDV-Neuling in der Lage sein, damit umzugehen.
Leider wird der positive Eindruck, den das Handbuch bietet, durch ein unvollständiges Register getrübt. Allzu oft ist man gezwungen, das Inhaltsverzeichnis zu Hilfe zu nehmen und auf gut Glück die in Frage kommenden Abschnitte durchzublättern, weil die entsprechenden Begriffe nicht im Index auftauchen.
Ist man im Bereich der Textverarbeitung schon ein „alter Hase", dann wird man nach dem Lesen der siebzig Seiten für Fortgeschrittene und zwei bis drei Stunden des Probierens kaum jemals mehr einen Blick in das Handbuch werfen müssen: So einfach ist die Arbeit mit dem Programm.

3. Arbeitsweise

Die effektivste Vorgehensweise, um Texte mit dem Computer zu verarbeiten, besteht aus vier Schritten. Zunächst gibt man den Text ein, ohne sich darum zu kümmern, wie die spätere Form aussehen soll, ob also z.B. manche Absätze eingerückt werden müssen.
Als zweites wird das Geschriebene auf logische Konsistenz, Stil und Tippfehler hin untersucht: Man redigiert den Text.
Schließlich legt man das äußere Erscheinungsbild des Druckes fest, indem etwa wichtige Begriffe unterstrichen oder durch Fettdruck hervorgehoben werden; auch das erwähnte Einrücken bestimmter Abschnitte gehört dazu. Im letzten Schritt überprüft man das Layout noch einmal und nimmt die notwendigen Korrekturen vor.
In der so vorgezeichneten Reihenfolge soll nun auch das Programm mit seinen zahlreichen Wahlmöglichkeiten vorgestellt werden.

3.1. Texteingabe

Mousewrite dürfte keine Schwierigkeiten haben, auch beim schnellen Zehnfingersystem mitzuhalten: Das Programm verkraftet mehr als zweihundert Anschläge pro Minute, ohne daß Buchstaben verlorengehen – und zwar unabhängig davon, ob sich der Cursor (das ist der Schreibzeiger) oben oder unten im Bild befindet. Manch andere Programme werfen schon bei sechzig Anschlägen pro Minute das Handtuch, wenn der Cursor am oberen Rand des Monitors steht.
Mitunter bewegt er sich allerdings für ein oder zwei Sekunden nicht von der Stelle, um dann sofort die inzwischen getippten Buchstaben zu präsentieren. Die Eingabe geht lautlos vor sich; man kann hier nicht wie bei einigen anderen Programmen ein „Klicken" einschalten, das den Anschlag der Schreibmaschinentypen auf dem Papier nachahmt.
Die Länge des maximal zu verarbeitenden Textes ergab sich als etwa 45.000 Buchstaben – das entspricht ca. 22 Schreibmaschinenseiten. Geht es auf die 45.000 Zeichen zu, treten mitunter störende Effekte auf, von denen noch die Rede sein wird.

3.2. Redigieren

Selbstverständlich sind alle üblichen Funktionen verfügbar: Einfügen von Text; Löschen einzelner Buchstaben, Wörter oder Abschnitte; Versetzen und Vervielfältigen von Textteilen; schließlich auch Finden und Ersetzen bestimmter Begriffe oder Zeichenkombinationen. Besonders die letzte Funktion ist sehr komfortabel ausgeführt. Will man etwa an einigen Stellen, an denen man „Hans" geschrieben hat, „Hans" durch „Franz" ersetzen, geht das Programm folgendermaßen vor: Es sucht den nächsten Ort, an dem „Hans" auftaucht, und zeigt diesen Abschnitt. Dann kann man sich jedesmal entscheiden, ob dieser „Hans" durch „Franz" ersetzt werden soll oder nicht, ob man im restlichen Text alle „Hans" umwandeln will, oder ob die Suche abgebrochen werden kann.

Läßt man Zeile für Zeile nach oben oder unten verschieben („scrollen"), fällt eine unzureichende Bewegungsgeschwindigkeit des Textes auf: Pro Sekunde erscheinen nur drei neue Zeilen. Für ein geordnetes Arbeiten ist das zu wenig; daher wird man auf eine andere Steuerungsart zurückgreifen müssen. Der rechte Seitenbalken des „Fensters", in dem der Text erscheint, dient als Hilfsmittel zum gezielten Anwählen bestimmter Dokumententeile. Führt man den Cursor mit Hilfe der Maus auf die Mitte des Balkens, sieht man die mittlere Stelle des Textes - unabhängig von seiner Gesamtlänge. Macht man dasselbe am unteren Ende des Balkens, wird das Ende gezeigt, usw..
Mitunter kann es nützlich sein, gleichzeitig zwei verschiedene Teile des Dokuments zu sehen, um z.B. den Gedankengang in diesen Abschnitten miteinander zu vergleichen. Zu diesem Zweck öffnet man ein weiteres „Fenster" zum Text. Die Größe der beiden „Fenster" kann beliebig festgelegt und ihr Inhalt unabhängig voneinander verschoben und redigiert werden.
Da man nur etwa 22 Seiten lange Dokumente bearbeiten kann, muß z.B. ein Buchmanuskript in einzelne Abschnitte zerlegt werden. Entscheidet man sich dann, daß ein bestimmter Absatz besser in das fünfte als in das achte Kapitel paßt, geht man wie folgt vor: Anfang und Ende des gewünschten Stücks aus dem achten Kapitel werden mit dem Cursor markiert; dann speichert man diesen Teil auf Diskette ab. Wird anschließend das fünfte Kapitel geladen, kann man den markierten Absatz an einer beliebigen Stelle einfügen.

4. Layout

Die Höhe des verwendeten Papiers und der Platz, der oben, unten und auf den Seiten freibleiben soll, wird vom Anwender in Zoll (= ca. 2,5 cm) angegeben. Ohne die großen Umstände, die das bei manchen anderen Programmen verur-

sacht, kann bei Mousewrite auch eine Kopf- und eine Fußzeile gedruckt werden, die auf jedem Blatt (außer der Titelseite) erscheinen.

In einer der beiden Zeilen können auch Seitenzahl, Datum und Uhrzeit erscheinen. Für Anwender, die keine Uhrenplatine im Apple eingebaut haben, bietet das Programm eine Software-Uhr an. Zu Beginn kann man Datum und Uhrzeit eingeben, die dann auch beim Abspeichern im Inhaltsverzeichnis der Diskette festgehalten werden. Allzu genau ist diese Software-Uhr natürlich nicht: Pro Stunde geht sie etwa zehn Minuten nach.

Der Text kann für den Druck linksbündig, rechtsbündig oder mittelachsig angeordnet werden, ebenso ist Blocksatz möglich; wie bei der Mehrzahl der Textverarbeitungsprogramme wird für Proportionalschrift kein Blocksatz unterstützt. Auf dem Bildschirm sichtbar ist das Layout jedoch nicht; statt dessen erscheint an einer Stelle, an der eine Veränderung der Seitenformatierung vorgenommen wird, eine grafische Marke.

Soll später im Text ein Bild eingefügt werden, das z.B. halb so breit wie das Blatt ist, muß der entsprechende Abschnitt eingerückt werden. Bei Mousewrite besteht nur die Möglichkeit, einen Absatz nach rechts einzurücken; Bilder können hier also immer nur am linken Rand eingefügt werden, niemals rechts.

Wenn Datumsangaben oder andere Textstücke im Dokument vorkommen, die nicht auf zwei Zeilen zerrissen werden dürfen, wird bei vielen Programmen so vorgegangen: Man schreibt nicht „1. 1. 1986", sondern z.B. „1. ↑ 1. ↑ 1986". Das Programm gibt statt der Pfeile Leerzeichen an den Drucker weiter und weiß, daß das zusammengehörige Textstück nicht am Zeilenende getrennt werden darf. Mousewrite bietet diese Möglichkeit nicht. Wenn man Pech hat, kann also der „1. 1. 1986" folgendermaßen auf dem Papier erscheinen:

„...das Treffen am 1.
1. 1986 verlief..."

Ein weiterer Schwachpunkt des Layout ist die Tatsache, daß *keine* Möglichkeit besteht, eine Silbentrennung durchzuführen, weder voll-, noch halbautomatisch. Wenn man ein langes Wort also trennen möchte, weil sonst eine zu große Lücke entsteht, muß das von Hand gemacht werden – ein Umstand, den man bei ernstzunehmender Textverarbeitungssoftware nicht hinnehmen kann.

Auch daß eine Absatzüberschrift allein unten auf einer Seite zu stehen kommt, während der zugehörige Abschnitt erst auf dem nächsten Blatt beginnt, kann bei Mousewrite nur durch manuelles Eingreifen verhindert werden – etwa, indem man vor der Überschrift eine neue Seite anfängt.

5. Drucken

Matrixdrucker bieten für gewöhnlich eine gewisse Auswahl an unterschiedlichen Laufweiten, etwa Pica, Elite oder Schmalschrift. Da alle Drucker aber verschieden bedient werden wollen – bei dem einen wird z. B. Schmalschrift durch „Escape Q", bei dem anderen durch „Shift In" ausgelöst –, hat das Programm für jeden von acht Druckern verschiedene Wertetabellen gespeichert.

Acht Auszeichnungsformen stehen zur Verfügung – falls der Drucker das auch alles zu Papier bringen kann: Unterstreichen, Fett-, Schmal- und Breitschrift, Exponenten, Indizes und zwei weitere Formen, die von Drucker zu Drucker verschieden belegt sind, z.B. mit Proportionalschrift.

Die entsprechenden Werte sind, wie gesagt, für acht Drucker bereits vorgegeben. Besitzt man etwa einen Apple Imagewriter, braucht man dies dem Programm nur mitzuteilen und kann sofort mit dem Drucken beginnen. Leider zeigt sich, daß nicht alle Voreinstellungen richtig getroffen wurden.

Im Handbuch ist z.B. angegeben, daß die erste „Extraform" beim Imagewriter Proportionalschrift sei. Bringt man einen entsprechenden Text zu Papier, stellt man jedoch fest, daß nicht proportional, sondern normal gedruckt wird. Also muß man sich die entsprechende Wertetabelle doch selbst zusammenstellen, was mit Hilfe des Druckerhandbuches allerdings nur wenige Minuten dauert.

Auch der Wechsel von einer Buchstabenform zur anderen wird auf dem Bildschirm durch ein grafisches Markierungszeichen dargestellt. Hat man an den Computer zwei verschiedene Drucker angeschlossen – z.B. einen Matrix- und einen Typenraddrucker –, braucht man vor dem Druckvorgang nur anzugeben, auf welchem der beiden die Ausgabe erfolgen soll, und kann dann mit allen verschiedenen Typenformen arbeiten, sofern sie vom Drucker unterstützt werden.

Wie nicht anders zu erwarten , kommt das Programm nicht damit zurecht, wenn man auf einer Zeile zwei Buchstabenformen mit verschiedener Breite verwendet. Mousewrite berechnet dann die Zeilenlänge so, als handle es sich um Zeichen gewöhnlicher Breite, und ruiniert dadurch den eventuell eingestellten Blocksatz oder rechten Randausgleich.

6. Bedienung

Zunächst wird es für die meisten Anwender ungewohnt sein, alle vorhandenen Funktionen mit Hilfe der Maus zu bedienen. Schon nach ein oder zwei Stunden Arbeit mit dem Programm geht man damit aber wie selbstverständlich um. Die Präsentation der verschiedenen Optionen – wie Randausgleich, Buchstabenform etc. – geschieht so übersichtlich, daß keine Verwirrung ob der großen Zahl von Wahlmöglichkeiten aufkommen kann.

Ist man dann mit dem Programm vertraut, wird man in vielen Fällen seine Wahl mit Hilfe der Tastatur vornehmen wollen, um nicht zur Maus greifen zu müssen. Die Offener-Apfel- und Geschlossener-Apfel-Tasten dienen zur Funktionswahl: Möchte man z.B. einen Satz unterstreichen, drückt man „Offener-Apfel U", und das entsprechende Markierungszeichen erscheint im Text. Manche Dinge gehen aber mit der Maus einfacher vonstatten, etwa das Verschieben oder Kopieren von Absätzen.

Druckt man einen langen Text aus, der aus mehreren Dokumenten besteht – z.B. ein Buchmanuskript –, dann ist man gezwungen, nach kurzen Pausen immer wieder zum Drucker zurückzukehren. Man kann dem Programm nicht etwa eine Liste der Texte geben, die nacheinander zu Papier

Softbreaker 1.0

Eine softwaremäßige Interrupt-Utility für die Apple IIe 64K-Karte

von U. Stiehl

1984, Diskette und Manual, DM 20,–
ISBN 3-7785-1022-3

Produkt läuft aus. Ab 1. 6. 86 Diskette mit Quellcode für nur noch DM 20,–

Softbreaker ist ein Assemblerprogramm, mit dessen Hilfe Programme, die sich von der 64K-Karte (= Extended 80 Column Card für den Apple IIe) starten lassen, unterbrochen, gespeichert, geladen und exakt an der Stelle der Unterbrechung fortgeführt werden können. Dadurch ist es auch möglich, Sicherungskopien von sogenannten kopiergeschützten Programmen herzustellen.

Mit Softbreaker unterbrochene Programme werden komplett, d. h. die ganzen 64K einschließlich Language Card, in nur ca. 11 Sekunden auf einer formatierten Diskette gesichert.

Gerätevoraussetzung: Apple IIe mit 64K-Karte, nicht IIc, nicht neue ROMs

Hüthig Software Service,
Postfach 10 28 69, D-6900 Heidelberg

gebracht werden sollen; statt dessen lädt man das erste Dokument von der Diskette und läßt es ausdrucken, dann lädt man das zweite Dokument und druckt es, etc.. Wenn man auf diese Weise 200 Seiten Manuskript erhalten hat, dürfte man sich fragen, ob die Anschaffung von Mousewrite wirklich $ 125 Wert war, denn der Hersteller wirbt mit den Worten: „Leistungsfähige Textverarbeitung von der Aktennotiz bis zum Manuskript...".
Es wurde schon erwähnt, daß man Dokumente bis zu einer Länge von ca. 45.000 Zeichen verarbeiten kann. Es empfiehlt sich jedoch nicht, allzu nahe an diese magische Grenze heranzukommen. Beim ersten Versuch, die maximale Speicherkapazität zu ermitteln, rührte sich der Cursor plötzlich nicht mehr von der Stelle. Der Computer reagierte erst wieder, als Control-Reset gedrückt wurde: Der Text war noch wie zuvor, nur vier Zeilen in der Mitte des Bildschirms hatten sich wie durch Zauber in lauter „ü"s verwandelt. Bei einem späteren Versuch, den rechten Rand bei fast vollem Speicher zu verschieben, spielte der Computer wieder „Toter Mann", nur daß sich der Apple beim Control-Reset diesmal mit einem höflichen Piepsen vom Tester verabschiedete und in den Niederungen des Monitorprogramms versank. Eine Störung wie die „ü"s kann man gerade eben noch hinnehmen, weil man wohl meist in der Lage ist, den verlorenen Text aus dem Gedächtnis zu rekonstruieren. Was ein Anwender jedoch denkt, wenn sich die Arbeit von Stunden mit einem Piepsen in Wohlgefallen auflöst, dürfte nicht druckreif sein.

7. Fehlende Sonderfunktionen

Wer von Zeit zu Zeit Briefe mit fast gleichem Inhalt an viele verschiedene Adressaten verschicken muß, braucht ein Textverarbeitungsprogramm mit Serienbrieffähigkeit. Mousewrite muß in diesem Punkt zurückstecken; es können damit keine variablen Dokumente gedruckt werden.
Für viele Zeitschriftenautoren und Journalisten ist es wichtig zu wissen, wieviele Anschläge das geschriebene Dokument hat, da oft Artikel von einer bestimmten Länge gefordert werden. Die Zahl der Anschläge liefert Mousewrite nicht, wohl aber rechnet es aus, wieviele Wörter und Zeilen der Text hat, und in welche Schulklasse man gehen muß, um den Inhalt verstehen zu können. Dieser „Lesbarkeitsindex" wird unter anderem aus der durchschnittlichen Wortlänge berechnet; er hängt nachgewiesenermaßen tatsächlich mit der Verständlichkeit eines Textes zusammen. Im vorliegenden Fall wäre es wohl sinnvoller gewesen, die Zahl der Anschläge feststellen zu können.

Fazit

Mousewrite dürfte zur Zeit wohl das am angenehmsten zu bedienende Textverarbeitungsprogramm auf dem Apple-Markt sein. Auch der Autor konnte sich diesem speziellen Charme nicht entziehen. Für viele Anwender kommt es sicherlich wegen der beschriebenen Schwächen nicht in Frage. Wer aber keine Serienbriefe schreiben will, wer mit den genannten Layout-Problemen leben kann, und wer sich lieber vom Computer bedienen läßt als den Computer zu bedienen, wird mit Mousewrite rundum glücklich werden.

Ein-Blick

Name	Mousewrite
Notwendige Ausstattung	Apple IIc / IIe mit erweiterter 8Ø-Zeichen-karte und neuen Zeichensatz-ROMs, 1 Diskettenlaufwerk (besser 2), Monitor, Drucker
Betriebssystem	ProDOS
Einsatz	Textverarbeitungsprogramm
Gesamtwertung	??? (Standardfunktionen fehlen)
Dokumentation	**** (23Ø Seiten)
Zweckdienlichkeit	??? (Standardfunktionen fehlen)
Optischer Eindruck	*****
Allgemeine Bedienbarkeit	*****
Verhalten bei Bedienungsfehlern	****
Preis-Leistungs-Verhältnis	??? (Standardfunktionen fehlen)
Kopierbar	Ja
Preis	$125.ØØ
Demonstrationsdiskette	$3.ØØ
Bezugsquelle	Roger Wagner Publishing, Santee, USA
Bewertungsschlüssel:	***** Hervorragend **** Überdurchschnittlich *** Akzeptabel ** Schlecht * Katastrophal

---

Hardware-Kurztip

Anschluß eines Brother HR-15 XL II an den Apple IIc

von Hans-Peter Sprauer

Ich bin Besitzer eines Apple IIc und eines Typenraddruckers vom Typ Brother HR-15 XL II. Damit stand ich vor dem Problem, diese beiden Geräte miteinander zu verbinden. Hier meine Lösung:
Benötigt werden für den seriellen Ausgang des IIc ein DIN 5poliger Diodenstecker und für den Brother ein 25poliger Min-D-Stecker. Als Verbindung reicht ein dreiadriges Litzenkabel. Nun geht es an das Verbinden (Beachten Sie dabei bitte die nachfolgende Abbildung):

```
   Apple //c                Brother
    Stift 5  ------------ Stift 2Ø
(data set ready)  (data terminal ready)
    Stift 2  ------------ Stift  3
(transmitted data)      (received data)
    Stift 3  ------------ Stift  7
(signal ground)         (signal ground)
```

Zusätzlich müssen im 25poligen Min-D-Stecker folgende Stifte überbrückt werden:
Stift 4 (ready to send) mit Stift 5 (clear to send) und Stift 6 (data set ready) mit Stift 8 (received line signal detector).
Die serielle Schnittstelle muß auf 7 Datenbits und 1 Stoppbit eingestellt werden. Zwar verarbeitet der Brother auch 8-Bit-Daten, verwendet dabei jedoch den IBM-Zeichensatz. Die Einstellung von Zeilenvorschub, Parität und Übertragungsrate bleibt dem Anwender überlassen, da der Brother hier über die gleichen Einstellungsmöglichkeiten wie der Apple verfügt. Zusätzlich kann man noch eine Zeilenlänge von 132 Zeichen einstellen.
Wollen Sie einen anderen Drucker als den Brother an den Apple IIc anschließen, empfiehlt sich ein Blick ins Handbuch dieses Druckers. Viele Herstellerfirmen verwenden nämlich andere Signalleitungen als Stift 4, 5, 6 und 8.

Ich hoffe, dieser Bericht hilft anderen Anwendern, ihren Apple IIc an ihren ganz persönlichen Drucker anzuschließen.

Bücher

Apple DOS unter der Lupe

von D. Worth und P. Lechner
1985, 202 S., kart., DM 32,-
Birkhäuser Verlag, Basel

Gliederung
Die Entwicklung des DOS – Die Formatierung einer Diskette – Die Diskettenorganisation – Der Aufbau des DOS – Der Gebrauch des DOS aus dem Assembler – Die Anpassung des DOS – Das DOS-Programm

Bemerkungen
Dieses Buch versteht sich als Ergänzung zum Apple-DOS-Handbuch, dessen Inhalt die Autoren voraussetzen. Dabei bietet das Buch, das sich auf die ursprüngliche Version DOS 3.3. bezieht, dem Apple-Neuling grundlegende Informationen über die Struktur von Disketten und dem fortgeschrittenen Assemblerprogrammierer Hinweise zur Programmierung eigener Diskettenzugriffsroutinen. Im Anhang werden 5 Beispielprogramme zur Untersuchung und Reparatur von Disketten aufgelistet.

Apple II für Technik und Wissenschaft

von R. Drewes
1984, 268 S., kart., DM 49,-
Data Becker Verlag, Düsseldorf

Gliederung
Programmieren in Applesoft-BASIC – Datenein- und -ausgabe – Anwenderprogramme aus der Mathematik, Chemie, Physik, Atomphysik und Technik – Programme zu Graphischen Darstellungen – Programme zur Datenverarbeitung

Bemerkungen
Dieses Buch enthält eine Vielzahl von BASIC-Programmen zur Lösung naturwissenschaftlicher Aufgabenstellungen. Dabei werden die zugrundeliegenden Gesetze und Formeln allenfalls kurz dargestellt, der Schwerpunkt des Buches liegt eindeutig auf der Wiedergabe der zahlreichen Listings.

Das Buch zum Apple-Writer II/IIe

von H. Gabriel
1986, 157 S., kart., DM 35,-
Dr. Alfred Hüthig Verlag, Heidelberg

Gliederung
Allgemeine Einführung – Die Apple-Writer Befehle – Die Programm-Steuerung – Tips und Tricks – WPL(TKS)-Programm analysen – Anhang

Bemerkungen
Diese Buch wendet sich an alle Benutzer dieses Textverarbeitungssystems, die mit einem Apple II+, IIc oder IIe arbeiten. Der Leser erlernt alle Funktionen und Befehle des Apple-Writers und wird im 3. Kapitel in die Programmierung des Apple-Writers in Text-Kommando-Sprache (TKS) eingeführt. Zusätzlich zum Buch kann eine Begleitdiskette zum Preis von DM 44,- bezogen werden, die alle Beispielprogramme, Schwerpunkterklärungen (über Help-Funktionen) und eine kleine Adreßdatenbank mit Programmen zur Datenverwaltung enthält.

Schulsoftware-Katalog Apple

Verzeichnis privater und kommerzieller Anwendungsprogramme für den Schulbereich
1985, 148 S., kart., DM 18,-
Ernst Klett Verlag, Stuttgart

Gliederung
Programmbeschreibungen zu einzelnen Sachgebieten: Informatik – Mathematik – Physik – Verwaltung – Biologie – Chemie – Sprachen – Wirtschaftswissenschaften – Sonstiges – Informationen zur Firma Apple – Der Einsatz des Apple II in der Schule – Händler- und Herstellerverzeichnis – Programmverzeichnis für den Macintosh

Bemerkungen
Der Katalog versteht sich als Softwarebörse für Anwender im Schulbereich. Die innerhalb von Sachgebieten alphabetisch geordneten Programme werden inhaltlich beschrieben, Hinweise zum Unterrichtseinsatz, zur Hard- und Software, über Gebühren und Anbieteradressen runden den Überblick ab.

Computer für Kinder

Ausgabe Apple II/IIe
von S. Greenwood Larsen
1984, 95 S., kart., DM 29,50
te-wi Verlag, München

Gliederung
Was ist ein Computer? – Flußdiagramme – Dein Computer – Programme speichern – Wichtige BASIC-Befehle – Grafikprogramme – Beispielprogramme – Verzeichnis der Befehle – Hinweise für Lehrer und Eltern

Bemerkungen
Die Autorin will mit ihrem Buch Kindern, ihren Lehrern und Eltern die erste Begegnung mit dem Computer ermöglichen und auf kindgerechte Weise die Funktionsweise eines Computers und den Einstieg ins eigene Programmieren erklären. Das Buch richtet sich an Kinder im Alter von 8 bis 13 Jahren und ist auch für Commodore C64 und VC20 erhältlich.

Graphik- und Elektronikprogramme für den Apple II

von H.M. Berlin
1985, 184 S., kart., DM 39,80
beam-Verlag, Marburg

Gliederung
Allgemeines – Graphische Darstellungen – Signalanalyse – Statistik und Daten-Analyse – Passive Netzwerke – Aktive Filter – Halbleiter-Bauelemente – Zusätzliche Routinen

Bemerkungen
Dieses Buch ist als Einführung in die Themenbereiche Computergrafik und Elektronik für Ingenieure, Techniker und Elektroniker sowie als Handbuch und Nachschlagewerk gedacht. Für immer wiederkehrende mathematische Aufgaben werden über 30 Lösungsvorschläge in Form von BASIC-Programmen für den Apple II, II+ und IIe vorgestellt, die allerdings einige mathematische Kenntnisse voraussetzen.

Datenmanagement mit dem Apple II

Ein BASIC-Programmpaket zum persönlichen Informationsmanagement
von N. Wadsworth
1985, 125 S., kart., DM 38,-
Vieweg Verlag, Braunschweig

Gliederung
Einführung in die computergestützte Datenbankverwaltung – Die Datenbankverwaltung – Bedienung des Datenbankprogramms – Anwendungen – Anatomie des Programms – Kommentar zum Programm – Hinweise für den Programmierer

Bemerkungen
Der Anfänger kann aus dem BASIC-Programmpaket das Programm zur Datenverwaltung einsetzen und erhält durch ausführliche Dokumentation und Erläuterungen zum Programm eine praktische Anleitung und Einführung in die Programmierung mit BASIC. Der Hobbyprogrammierer mit Erfahrung in BASIC kann Teile des Programms seinen Wünschen entsprechend verändern, weglassen, neu programmieren, da das Programmpaket als „offene Software" konzipiert ist.
Das Programmpaket ist auf Diskette erhältlich und kostet DM 38,–. Es kann über die angeheftete Bestellkarte bezogen werden.

Einführung in die Apple-UCSD-Pascal-Systeme

Eine Handreichung für die Schule und den privaten Benutzer
von A. Burkhardt
1985, 43 S., kart., DM 8,80
Biodidac-Verlag, Rheinbrohl

Gliederung
Allgemeines – Die System-Hauptebene – Der Editor – Der Filer – Der Compiler – Die Linker- und Assembler-Ebene – Die Programmebene – Praxis-Anhang

Bemerkungen
Diese Handreichung entstand aus dem Computerunterricht und soll als erste kurze, einfache, preiswerte und deutsch verfaßte Einführung ins System und als Nachschlagwerk für Neueinsteiger dienen. Sie will und kann kein Handbuch-Ersatz sein, spricht aber alle wichtigen Programmier- und Speicherarbeiten an. Im Unterricht soll der Anwender sein Arbeitsbuch selbst bearbeiten und Kommentare, eigene Erläuterungen etc. auf den freien Rückseiten vermerken. Um einen günstigen Preis zu ermögli-

chen, wurde das Arbeitsbuch im DIN A4-Format als Kopie eines Ausdruckes mit einem Typenraddrucker vervielfältigt.

The Custom Apple & Other Mysteries

von W. Hofacker und E. Floegel
1983, 190 S., kart., $24.95
IJG Inc., Upland California
In Deutschland Auslieferung über Hofacker Verlag, Holzkirchen

Gliederung
Allgemeines – Datenein- und -ausgabe – Tonerzeugung – 8-Bit-Digital-Analog-Wandler – EPROM-Programmiergerät – EPROM/RAM Board – Slot-Übertragungsleitung – Verbindung zweier 6502-Systeme – Verbindung mit anderen Mikroprozessoren – Kontrolle der Schrittmotoren

Bemerkungen
Englischsprachiges Paperback mit ausführlichen Hard- und Softwareinformationen.

EDV-kein Geheimnis

Grundwissen für Mitarbeiter und Chefs
von D.A. Schilling
1985, 118 S., kart., DM 36,-
Dr. Alfred Hüthig Verlag, Heidelberg

Gliederung
Einführung – Die Hardware – Die Software –Der Mikrocomputer, ein Spielzeug? – Die EDV in der Buchhaltung – Die EDV im Handwerk – Die Datenübertragung in der EDV – Wie sicher sind die Daten?

Bemerkungen
Aufgrund seiner einfachen Ausdrucksweise und der zahlreichen Skizzen und Beispiele spricht das Buch sowohl den Anfänger als auch Anwender und Praktiker sowie Mikro- und Homecomputer-Begeisterte an. Es will Lücken schließen, die bei der üblichen Schnellausbildung im Rahmen der EDV-Einführung für Mitarbeiter und Chefs unweigerlich entstehen und bleiben. Die verständliche Art der Darstellung basiert auf der über 10jährigen Erfahrung des Autors als EDV-Instruktor am Schulungszentrum eines EDV-Herstellers.

Spielend Programmieren lernen

von K.-H. Koch
1985, 159 S., kart., DM 8,80
Humboldt-Taschenbuchverlag, München

Gliederung
Es ist leichter, als Sie denken – BASIC-Basis – Farbenspiele – Spielplätze – Mehr Spiele – Programmierhilfen – Tabellen

Bemerkungen
Der Autor will dem Leser das Programmieren in BASIC nicht anhand von Lehrbeispielen aus der Wirtschaft, sondern an Spielprogrammen vermitteln. Sinn des Lehrbuches ist also das fast mühelose, spielerische Erlernen der Programmiersprache BASIC und das Computerspiel, dessen einzelne Programmierschritte für die gängigsten Computertypen gut vorbreitet und erklärt werden. Daneben findet auch der Freak noch ein paar Tips und Tricks, um seinen Programmaufbau zu vereinfachen.

Programmiersprache BASIC – Schritt für Schritt

Systematische Darstellung der Computersprache, die am einfachsten zu erlernen ist. Viele Programmbeispiele für die Praxis.
von Dr. H.-J. Sacht
1983, 160 S., kart., DM 7,80
Humboldt-Taschenbuchverlag, München

Gliederung
Computer und Computersprachen – BASIC als Dialogsprache – Den Computer bedienen lernen – Zahlendarstellung und Rechnen – Variable und symbolische Namen – Nur 4 Grundbefehle sind notwendig – Fehler und Fehlersuche – Die Darstellung auf dem Bildschirm – Numerische Funktionen – Weitere Befehle – Fehler erleichtern das Programmieren – Zeichenkettenbearbeitung – Speichern und Drucken – Einige Befehle für Fortgeschrittene – Das Vorgehen bei der Programmierung

BASIC für Fortgeschrittene

24 praktische Anwenderprogramme –
Für Leute mit Anfangskenntnissen in BASIC
von Dr. H.-J. Sacht
1984, 191 S., kart., DM 8,80
Humboldt-Taschenbuchverlag, München

Gliederung
Was Sie schon über BASIC wissen sollten – Abfragen, Entscheiden und Schleifen – Aufbau und Entwicklung von BASIC-Programmen – Beispielprogramme für reinen Bildschirmbetrieb – Verwendung von Druckern – Beispielprogramme mit Druckerausgabe – Datenspeicherung in sequentiellen Dateien – Beispielprogramme mit Datenspeicherung – Dateien mit wahlfreiem Zugriff – Beispielprogramm für wahlfrei lesbare Dateien

Das Hacker HACKBUCH

von R. Bouteiller
1985, 130 S., kart., DM 19,80
edition aragon, Verlag W. Klauke, Moers

Gliederung
Die Robin Hoods der 80er? – Die Eintrittskarte – Hardware – Grundlagen der Datenfernübertragung – Die Datennetze – Kein Eintritt ohne diese Nummer – Die elektronischen Briefkästen – Bildschirmtext – Anhang: Mailbox- und DATEX-P-Rufnummern, Adressen von Datenbankanbietern, Internationale Netzkennzahlen, Akustikkoppler –Geräte und Adressen

Bemerkung
Ein locker geschriebenes Handbuch für Hacker und alle, die es werden wollen.

Mikrowissen A-Z

Mit 950 Stichworten und zahlreichen Abbildungen
Register Deutsch-Englisch und Englisch-Deutsch
von G. Rolle
1985, 124 S., kart., DM 24,80
Vieweg Verlag, Braunschweig

Bemerkungen
Mikrowissen A-Z erklärt die wichtigsten Hard- und Software-Fachbegriffe aus dem Bereich der Home- und Personal-Computer. Eine Reihe von Abbildungen und Tabellen unterstützt die Begriffserklärungen. Das Register Englisch-Deutsch/Deutsch-Englisch dient dem besseren Verständnis englischsprachiger Texte.

Technisch-naturwissenschaftlicher Pascal-Trainer

von H. Kohler
1985, 218 S., kart., DM 38,-
Vieweg Verlag, Braunschweig

Gliederung
Einleitung – Struktogramme – Verzweigungen und schleifenfreie Programmierung – Fehler und Schleifen, Sprünge und Verzweigungen – Unterabläufe – Aufzähl- und Mengen-Typ (SET) – Verbund-Typ (RECORD) – Dateien und Texte – Pointer-Variable (Zeiger-Typ) – Die Aufgabe „ohne Netz" – Lösungen

Bemerkungen
Das Buch vermittelt Anleitungen und Anregungen für die Programmiersprache Pascal durch Übungen verschiedenen Schwierigkeitsgrades und ihre Lösungen. Dabei wurde versucht, im Rahmen des Pascal-Standards möglichst viele Sprachvariationen in die Lösungsvorschläge einzuarbeiten.

Computereien

Heiteres aus der Welt der Computer und Elektronik
Hrsg. H.-A. Herchen
Bd. 1, 1984, 2. Auflage, 96 S., DM 8,80
Bd. 3, 1984, 96 S., DM 8,80
Haag+Herchen Verlag, Frankfurt

Bemerkungen
Die beiden Bände enthalten zu den genannten Themenbereichen heitere Begebenheiten, Witze, Aphorismen, Computertricks, „computerisierte" Erzählungen und Karikaturen.

Computereien 2

Cartoons aus der Welt der Computer und Elektronik
von W. Hasenpusch
1983, 96 S., kart., DM 8,80
Haag+Herchen Verlag, Frankfurt

Bemerkungen
Eine Sammlung amüsanter und witziger Karikaturen, in deren Mittelpunkt der Computer steht.

Auswahl und Einsatz lokaler Netzwerke

Neue Wege der Informationsbewältigung
von U. Gollub und D. Ahlers
1985, 115 S., kart., DM 34,-
Dr. Alfred Hüthig Verlag, Heidelberg

Gliederung
Einleitung – Allgemeine Probleme der Informationsflußanalyse – Theoretische Grundlagen der Lokalen Netzwerke – Anforderungen an ein Lokales Netzwerk – Kostenbetrachtung – Wirtschaftlichkeitsbetrachtung – Organisationsergonomie – Informationsfluß-Ingenieur – Einführungsstrategie für Lokale Netzwerke – Ausblick
Bemerkungen
Im diesem Buch werden Möglichkeiten aufgezeigt, mit sog. „Autonomen Informationsinseln" – das sind Subsysteme innerhalb von Unternehmen, die alle für sie relevanten Daten und Verarbeitungssysteme beinhalten und gruppenspezifisch aufbereiten – große, vollintegrierte, relativ unüberschaubare Netzwerke ökonomisch zu ersetzen.

FORTH ganz einfach

von T. Hogan
1985, 77 S., kart., DM 29,80
Vieweg Verlag, Braunschweig
Gliederung
Was ist FORTH? – FORTH ist anders – Der FORTH-Stapel – Rechnen mit FORTH – Die Anwendung – FORTH als Interpreter und Compiler – Speichermanipulationen – Logische Verknüpfungen – Kontrollstrukturen – Ein- und Ausgabe – Programmieren in FORTH
Bemerkungen
Für den Programmieranfänger und den Fachmann erklärt T. Hogan ausführlich die FORTH-Syntax – besondere Berücksichtigung finden die Dialekte FORTH-79 und FIG-FORTH und Hinweise für logische Erweiterungen und Alternativen dieser Dialekte im Vergleich zum FORTH-Standard. Durch die lockerere Sprache und die reich bebilderte Darstellungsweise wird dem Leser das Verständnis erleichtert.

Computerspiele und Computergrafik

Anleitung zur eigenen Software-Entwicklung
von K. S. Reid-Green
1986, 314 S., kart., DM 50,-
Vieweg Verlag, Braunschweig
Gliederung
Einführung – Das Programmierrüstzeug – Die Bedeutung des Zufalls – Realitätsnähe und Lebendigkeit – Ballistische Spiele – Rennbahnspiele – Irrgarten- und Phantasiespiele – Weltraumspiele – Lehr- und Lernspiele – Computer Aided Design – Ideen und Spiele für Hobbyprogrammierer – Anhang
Bemerkungen
Ziel des Buches ist es, den Leser beim Schreiben eigener Strategie- und Spielprogramme zu unterstützen. Der Schwerpunkt liegt auf der Erklärung und Beschreibung der Problemstellung, den Anfangsbedingungen und der Spielregeln. Alle Programme sind in strukturiertem BASIC (Pseudocode) geschrieben.

Programmieren in FORTRAN 77

von K. Brauer
1986, 307 S., kart., DM 46,-
Dr. Alfred Hüthig Verlag, Heidelberg
Gliederung
Einführung – Elemente von FORTRAN 77 – Arithmetische Operanden, Ausdrücke und Wertzuweisungen – Eingabe und Ausgabe von Daten – Einige Steueranweisungen – Logische Größen und Operatoren, Vergleichsausdrücke – Abfrage-Anweisungen – Arrays – Zeichenverarbeitung – Eingebaute Funktionen – Unterprogramme – Kommunikation zwischen Programmeinheiten – Vorbesetzung von Größen – Benutzung von Dateien, Ergänzungen zu Format-Spezifikationen – Weitere Möglichkeiten in FORTRAN 77
Bemerkungen
In diesem Buch werden alle Sprachmittel vollständig und systematisch beschrieben und durch vollständige, ausgetestete Beispiele erläutert. Detailliert wird auf die Einschränkungen des „subset", des verringerten Sprachmittelumfangs auf Mikrocomputern, eingegangen. Das Buch versetzt den Leser in die Lage, selbständig größere Probleme in FORTRAN 77 bearbeiten zu können.

SYBEX Mailbox Führer

von B. Hurth
1985, 251 S., kart., DM 14,80
SYBEX-Verlag, Düsseldorf
Gliederung
Voraussetzungen für Teilnahme am Mailboxen – Übersicht über in- und ausländische Mailboxen nach Alphabet und Telefonnummern – DATEX-P – DFÜ-Lexikon
Bemerkungen
Dieses Buch gibt eine Einführung in diese moderne Form der Telekommunikation und die nötige Ausrüstung und beschreibt die wichtigsten Mailboxen und ihre Dienste. Durch zahlreiche Tips erspart es dem Leser Telefongebühren und Zeit bei der Suche nach den benötigen Informationen.

Rechnerunterstütztes Konstruieren (CAD)

von H. Aurich, L. Franz und S. Schönfelder
1985, 270 S., kart., DM 58,-
Dr. Alfred Hüthig Verlag, Heidelberg
Gliederung
Einleitung – Voraussetzungen für den Einsatz der EDV in der technischen Vorbereitung – Grundlagen für die Anwendung von EDVA – Rationalisierung einzelner konstruktiver Tätigkeiten – Rationalisierung von Tätigkeitskomplexen des Konstruierens durch Dialogtechnik – Aufbau, Struktur und Nutzung der Programmkette INKO – Erfahrungen bei der Anwendung der EDVA in der Konstruktion – Tendenzen der Anwendung der EDV im rechnerunterstützten Konstruieren
Bemerkungen
Dieses Buch wendet sich an alle Nutzer und Ersteller von CAD-Systemen, sowohl an den Studenten zur Einarbeitung in den Themenbereich als auch an den tätigen Ingenieur, dem es dank des engen Praxisbezugs wertvolle Hinweise liefert. Zahlreiche Beispiele stammen aus dem Fachbereich Maschinenbau.
Hinweise zur Entwicklung von Software zum dialoggeführten Konstruieren – insbesondere über die Menütechnik – und zum rechnergestützten Optimieren gelten dem Programmierer. Ein umfangreiches Literaturverzeichnis rundet das Gesamtbild ab.

Computergrafische Experimente mit Pascal

Chaos und Ordnung in Dynamischen Systemen
von K.-H. Becker und M. Dörfler
1986, 319 S., kart., DM 42,-
Vieweg Verlag, Braunschweig
Gliederung
Forscher entdecken das Chaos – Bausteine für grafische Experimente – Pascal und die Apfelmännchen – Anhang: UCSD-Pascal: Editor, Compiler, Fehlermeldungen, Syntaxprogramme – Technische Hinweise
Bemerkungen
Das Buch wendet sich an alle, die über ein Computersystem verfügen und Spaß am Experimentieren mit Computergrafiken haben. Die Verknüpfung von Wissenschaftstheorie und Computeralgorithmen, die Synthese von mathematischer Theorie und Programmierpraxis in Pascal machen dieses Buch zu einem Novum auf dem Gebiet der Computerbücher. (UCSD-Pascal-Programme für Apple II, Version Apple Pascal 1.1, für Macintosh und IBM-PC.) Alle im Buch abgedruckten Programme können auch auf Diskette bezogen werden.

Programmierkurs mit Microsoft-BASIC

Band 1: Grundlagen
von W. Hagenmüller
1985, 188 S., kart., DM 38,-
Dr. Alfred Hüthig Verlag, Heidelberg

Gliederung
Vorbemerkungen – Erste Schritte – Grundlagen für verzweigte Programme – Programme mit Verzweigungen und Schleifen – Die wichtigsten Phasen der Programmerstellung – Weitere Möglichkeiten von M BASIC
Bemerkung
Das Buch wendet sich an Leser, die einen Einstieg in die Technik des Programmierens suchen oder die Programmiersprache Microsoft-BASIC kennenlernen wollen. Die Befehle und Schritte zur Programmerstellung werden mit steigendem Schwierigkeitsgrad angeordnet und durch Übungsaufgaben und Wiederholungsfragen (mit Lösungen) zu jedem Abschnitt vertieft.

Computer im Unterricht

Konsequenzen für den Bildungsauftrag der Schule
Dokumentation der Tagung am 25./26.10.1985 in der Sparkassenakademie Landshut
Hrsg. Studienkreis Schule-Wirtschaft, München
1986, 233 S., kart., o. Preis
Bezug über den Studienkreis Schule-Wirtschaft, Brienner Str.7, 8000 München 2
Gliederung
Grundsatzreferate – Die Informationstechnik in den einzelnen Schularten: Erfahrungen, neue Ansätze, Perspektiven – Der Computereinsatz in der Hauptschule – Realschule – Gymnasium
Bemerkungen
Die Veröffentlichung ist das Ergebnis einer bundesweiten Tagung mit ca. 200 Teilnehmern. Im Vordergrund standen die Themenbereiche: Qualifikationsanforderungen der Wirtschaft – Erfahrungen mit dem Computer in der Schule – Einschätzung aus der Sicht der Erziehungswissenschaftler. Das pädagogische Ziel ist ein verantwortlicher Einsatz des Computers im Unterricht und die fächerübergreifende Vorbereitung der Jugendlichen auf den Umgang mit den neuen Techniken.

Modelle der Wirklichkeit

Simulation dynamischer Systeme mit dem Mikrocomputer
von H. Rauch
1985, 210 S., kart., DM 29,80
Heinz Heise Verlag, Hannover
Gliederung
– Programme: Wachstumsfunktionen – Radioaktiver Zerfall – Räuber-Beute-Beziehung – Wachstum der Weltbevölkerung unter Berücksichtigung des Faktors Landwirtschaft – Wachstum auf begrenzter Fläche – Die Tsembaga in Neuguinea – DDT in unserer Umwelt – Bevölkerungspyramide – Forrester-Weltmodell – Kritik
– Entwicklung eigener Simulationsprogramme
– Die Programmierumgebung und Hinweise für andere Pascal-Versionen
Bemerkungen
Im ersten Teil des Buches, das leider nur in Matrixdruckerschrift gesetzt wurde, werden 9 dynamische Simulationsprogramme (in Pascal) dargestellt. Im zweiten Teil wird am Beispiel des Modells „Waldsterben" erläutert, wie der Leser eigene Simulationsprogramme schreiben kann. Hierzu hat der Autor ein Programmiersystem entwickelt, das die Simulationssprache DYNAMO in ihren wesentlichen Elementen auf dem Mikrocomputer nachbildet.
Zum Buch können für die Rechner Apple (mit Z80-Karte), Alphatronic PC, IBM PC und Kompatible und Schneider CPC Disketten erworben werden:
DI: 2 Programmdisketten, 9 Simulationsprogramme, DM 58,-
DII: 2 Entwicklungssytemdisketten zur Entwicklung von Simulationsprogrammen, DM 58,-

Ausgabe und **Eingabe** mit
TYPETERM®
im Slot Ihres
APPLE II/IIe

Das bedeutet: Computertextverarbeitung von der Schreibmaschinentastatur! Steckerfertig ohne Umbau.

Die neue CE-550!
mit TYPETERM DM 1.398,–

TYPETERM-Interface DM 479,–
für alle BROTHER-Typenradschreibmaschinen ab AX-30 bis EM-811
(auch für Vorgängermodelle!)
Paketpreis z. B.:

EM-501 mit TYPETERM ..... DM 2136,–
EM-511 mit TYPETERM ..... DM 2412,–
EM-701 mit TYPETERM ..... DM 2468,–

TYPETERM – ein starkes Interface für starke Maschinen! Alle Cursor- und Ctl-Befehle. 4k ROM auf der Karte für DOS, PRODOS, CP/M, PASCAL. 2 Zeichensätze verfügbar z. B. deutsch u. ASCII. Alle Features: Hoch-/Tiefstellen, autom. Unterstreichen, var. Zeichen und Zeilenabst., autom. Papierzuführung usw.
TYPETERM – ein Produkt von
interkom electronic Kock & Mreches GmbH Postf., 3004 Isernhagen 4 Telefon 0 51 39-8 73 93

Ausgabe mit
TYPETERM® JUNIOR
im Slot Ihres
APPLE II/IIe

Paketpreis **DM 899,–**
Schreibmaschine AX-10 mit Interface TYPETERM JUNIOR, steckfertig.

brother
Die Zukunft heute

TYPETERM JUNIOR mit AX-10 – unser besonders günstiges Gespann, ebenfalls steckfertig. Mit TYPETERM JUNIOR kann die AX-10 mehr. Sie wird zum vollwertigen Typenraddrucker für Ihren Apple:
- 3 verschiedene Schriftstärken
- Automatisches Unterstreichen
- 2 Zeichensätze z.B. deutsch u. ASCII
- 2 Zeichenabstände
- 2k ROM auf der Karte für Ausgabe unter DOS, PRODOS, CP/M u. PASCAL.

TYPETERM JUNIOR – ein Produkt von
interkom electronic Kock & Mreches GmbH Postf., 3004 Isernhagen 4 Telefon 0 51 39-8 73 93

Laser 128

Ein Apple-IIc-Kompatibler

getestet von Hans-Martin Eng

Kürzlich fragte mich Herr Stiehl, ob ich bereit sei, den Apple IIc-kompatiblen Laser 128 zu testen. Ich hatte schon einen (allerdings recht oberflächlichen) Test in einer bekannten deutschen Computerzeitschrift gelesen und war dadurch recht neugierig. Ich sagte also zu. Einige Tage danach rief Herr Stiehl an und teilte mir mit, daß das Testgerät eingetroffen sei: „Eine Riesenkiste", sagte er. Riesenkiste? Für einen IIc-Kompatiblen? Das machte mich noch neugieriger.

1. Lieferumfang

Der 57 x 57 x 47 cm große Karton enthielt neben einigem Füllmaterial:
- den Laser 128
- ein externes Laufwerk
- eine Maus
- je ein Kabel für den seriellen und parallelen Druckerausgang
- ein Netzteil
- ein Videokabel
- ein englischsprachiges Handbuch

Laut Lieferschein kostet das zusammen ca. DM 2600,-, ist also deutlich billiger als eine vergleichbare Konfiguration mit Originalgeräten.

2. Technischer Überblick

Der Laser 128 ist mit einer Grundfläche von ca. 37 x 32 cm deutlich größer als sein Vorbild. Er verfügt über eine Schreibmaschinentastatur mit abgesetztem Ziffernblock. Oben befindet sich noch eine Reihe mit 10 Funktionstasten, die jedoch fest mit Steuerzeichen belegt, also nicht programmierbar sind. Statt der Apfeltasten verfügt der Laser über Tasten mit Dreiecksymbolen. Diese sind jedoch ebenso mit den Druckknopfeingängen verbunden.
Auf der rechten Seite des Geräts ist ein Diskettenlaufwerk eingebaut, das im Gegensatz zu dem des Originals 40 Spuren lesen und schreiben kann. Auf der linken Seite ist ein Apple-kompatibler Slot herausgeführt. Die Rückseite weist die üblichen Anschlüsse auf, zusätzlich jedoch bietet der Laser einen Centronics-Parallelausgang.
Verschiedene Grundeinstellungen sind über vier Umschalter wählbar. So kann man zwischen 40- und potentieller 80-Zeichendarstellung wählen. Die Druckerschnittstelle wird über einen Umschalter selektiert, das Videosignal kann speziell auf Monochrom-Monitore umgestellt werden. An der Geräteunterseite schließlich besteht die Wahlmöglichkeit zwischen 50 und 60 Hz Bildfrequenz. Wählt man 50 Hz, erzeugt der Laser 128 PAL-Farbsignale, bei 60 Hz NTSC-Signale.
In meinem Testgerät war ein Z80-Prozessor zusätzlich zum 65C02 eingebaut. Damit war das Gerät voll für CP/M-2.2 tauglich. Ob dies jedoch zur Grundausstattung gehört, konnte ich aus dem Handbuch leider nicht erfahren. Einerseits war die Aufrüstung nicht im Lieferschein aufgeführt, andererseits weist ein beigelegter Prospekt auf ein getrennt erhältliches Z80-Modul hin.
Im ROM befindet sich ein Applesoft-kompatibler Microsoft-BASIC-Interpreter. Zusätzlich wurde ein komplettes Setup-Programm für die Schnittstellen implementiert. Der hierfür benötigte Speicherplatz geht jedoch voll zu Lasten des Monitors, der nur noch sehr eingeschränkte Funktionen anbietet. So sind der List-, der Verify- und der Examine-Befehl restlos gestrichen, ebenso wird keine Hexadezimalarithmetik mehr angeboten.

3. Funktionstest

Der angenehme äußere Eindruck der Tastatur wurde durch den Betriebstest doch ein wenig relativiert. Die Tasten klappern mir ein wenig zu laut, und der Druckpunkt wird sehr abrupt erreicht. Die Reset-Taste ist schlicht und ergreifend zu klein ausgefallen. Abgesehen vom unangenehmen Tastgefühl ist der Ziffernblock eindeutig als Gewinn anzusehen.
Der Umschalter für die Tastaturbelegung erregte bei mir jedoch nur Kopfschütteln. Damit schaltet man nämlich nicht auf eine DIN-Tastatur um, vielmehr wird die – auch für amerikanische Verhältnisse als äußerst exotisch einzustufende – Dvorak-Tastenbelegung eingeschaltet. Bei ihr ist zuerst einmal alles anders als auf den Tastenkappen aufgedruckt. Für Interessierte zeigt **Tab. 1** ein Schema dieser Belegung.
Die Qualität der Zeichendarstellung ließ an und für sich nichts zu wünschen übrig, die Bildposition auf dem Monitor war jedoch zu tief. Glücklicherweise verfügt mein Monitor über die vielfältigsten Einstellungsmöglichkeiten. Doch was sollen Besitzer anderer Fabrikate unternehmen? Laut Handbuch verfügt der Laser neben den Standard-Grafikmöglichkeiten noch über eine vierfach hochauflösende Grafik. Da sich das BASIC-Handbuch ansonsten aber darüber ausschwieg und mir weder ein Reference-Manual noch eine Diskette

Tabelle 1

Esc	1	2	3	4	5	6	7	8	9	0	–	=	Del
Tab	/	,	.	p	y	f	g	c	r	l	Ä	Ü	Ö
Ctrl	a	o	e	u	i	d	h	t	n	s	'	Return	
Shift	;	q	j	k	x	b	m	w	v	z	Shift		
Caps ß	off	Leertaste				ges	li	re	un	ob			
Lock	Apf					Apf							

Preiswerte Begleitdisketten

Bd. 1: DM 28,–; Bd. 2: DM 28,–

DM 28,–

DM 28,– (Neue Diskette für 3. Aufl.)

Hüthig Software Service
Postfach 10 28 69 · 6900 Heidelberg

mit Programmbeispielen zur Verfügung standen, konnte ich diese Grafikmöglichkeit leider nicht begutachten – schade.
Die Idee mit dem Slot gefiel mir zwar ganz gut, ihre Realisation läßt jedoch erheblich zu wünschen übrig. Die Kontakte des Slots liegen praktisch offen zu Tage. Wer ihnen beispielsweise mit einem Schlüssel zu nahe kommt, kann eine besondere Art von Reset erzeugen – und den Rechner anschließend zur Reparatur einschicken. Zumindest mit einem Blindstecker könnten die Hersteller ihren Laser 128 ausliefern.
Bei längerem Betrieb trat bei meinem Testgerät ein hochfrequentes Piepsen auf. Auf die Dauer ging mir dieses Geräusch ziemlich auf die Nerven. Eigentlich ist das ein Fall für die Qualitätskontrolle des Herstellers. Oder gehört das Piepsen etwa auch zu den Features des Laser 128? (Nach dem Motto: „Nie wird unser Laser 128 überhört werden, dazu ist er einfach zu aufdringlich!" – oder so ähnlich.)
Ein eindeutiger Pluspunkt sind jedoch die Laufwerke des Laser 128. Ihre Laufruhe übertrifft die meiner schon sehr leisen Siemens-Laufwerke noch einmal deutlich. Im Test traten keinerlei Probleme mit ihnen auf. Somit dürfte das externe Laufwerk ein heißer Anschaffungstip für Besitzer eines Original-Apple IIc sein.

4. Kompatibilität?

Ja, das Fragezeichen ist Absicht (und angebracht). Von der Hardware her gesehen ist der Laser 128 ja aufwärtskompatibel zum Apple IIc, doch wie steht es mit der Firmware? Ich deutete schon an, daß vor allem das Konfigurationsprogramm für die Schnittstellen eine Menge Speicherplatz frißt. So ging ich mit einem schlechten Gefühl an den Softwaretest heran.

4.1. DOS 3.3 und ProDOS

Zuerst ging alles gut. Reine BASIC-Programme liefen einwandfrei ab. ProDOS bootete, ohne einen Patch zu benötigen. Doch schon mein PLOT.3.E stieg bei der Funktionseingabe rettungslos aus. Das heißt, daß alle Programme, deren Assemblerroutinen auf die Applesoft-Routinen zurückgreifen, potentielle „Aussteiger" sind. Dies ist jedoch bei einem hohen Prozentsatz der unter DOS 3.3 und ProDOS erhältlichen kommerziellen Programme der Fall. Genauso übel machen sich die Änderungen im Monitor-ROM bemerkbar. Alle mir bekannten Disassemblerprogramme greifen beispielsweise auf die List-Routine des Monitors zurück – schlechte Karten für Assembler-Interessierte. Die Liste der mir bekannten, auf dem Laser 128 nicht einwandfrei laufenden Programme ist hier keineswegs vollständig wiedergegeben.

4.2. PASCAL und CP/M

Bei den anderen Betriebssystemen darf man jedoch aufatmen. Da diese nur in ganz geringem Maße auf ROM-Routinen zugreifen, laufen beide einwandfrei an. Ich konnte keine Probleme beim Testen feststellen, die Parallelschnittstelle wurde beispielsweise einwandfrei erkannt und angesteuert. Die CP/M-Standardprogramme Turbo-Pascal, Wordstar und Microsoft-BASIC80 liefen ohne Einschränkungen.

5. Das Handbuch

Das englischsprachige Handbuch gliedert sich in zwei Teile. Der eine beschreibt äußerst oberflächlich den Aufbau des Geräts, verweist im übrigen auf das „Reference-Manual", das mir leider nicht vorlag. Der andere Teil ist eine gut lesbare Einführung in die BASIC-Programmierung mit dem Laser 128.

6. Fazit

Offen gestanden hinterließ der Laser 128 bei mir einen sehr zwiespältigen Eindruck. Betrachtet man den niedrigen Preis und die Möglichkeit, CP/M direkt zu nutzen, so ist das Gerät sicherlich empfehlenswert. Schwer wiegen jedoch die Kompatibilitätsprobleme, die gerade noch zufriedenstellende Tastatur und der verunglückte Slot. Formulieren wir das Testergebnis so: Unter CP/M, UCSD-Pascal und (reinem) Applesoft-BASIC bedingt empfehlenswert, für Programme, die auf ROM-Routinen zurückgreifen, jedoch nur mit Abstrichen geeignet.

Keycom 6

Eine externe Tastatur für den Apple IIe

getestet von Hans-Martin Eng

Die Tastatur Keycom 6 ist auf einem Preh-Commander-Tastaturkörper aufgebaut. Die Anpassung an den Apple IIe besorgte die Firma SH electronic. (Anm.d.Red.: Eine Abbildung der Tastatur, die uns übrigens von der Firma Ueding in Menden ausgeliehen wurde, findet der Leser im Peeker, 5/86, S.6.)

1. Mechanischer Aufbau

Wer von der Firma Preh gehört hat, weiß, daß ihre Tastaturen auf Folienkontakten aufgebaut sind – keine Angst, das heißt nicht, daß man künftig auf einem Sinclair ZX-81-Verschnitt seine Eingaben erledigen muß. Vielmehr werden die Tasten gut geführt, besitzen ein angenehmes Rückstellmoment und einen gut fühlbaren Druckpunkt. Zusätzlich kann noch ein künstlicher Tastatur-Klick über einen eingebauten Lautsprecher ausgegeben werden.

Die Tastatur ist in einem formschönen, sehr flachen Gehäuse untergebracht, die Tastenkappen sind alle einheitlich grau, abgesehen von den Apfeltasten. Das Tastenfeld besteht aus drei Zonen, einem DIN-Schreibmaschinenblock, darüber angeordnet eine Zeile mit 15 Funktionstasten und rechts ein Ziffernblock.

Die Kodierung besorgt ein 6510-Prozessor, seine Betriebssoftware ist in einem 2764-EPROM untergebracht, die Tastenbelegung wird im batteriegepufferten RAM abgespeichert. Die Daten werden als 7-Bit-ASCII-Daten an den Apple geschickt, vom Apple kommen eine Handshakeleitung und eine Leitung zum Annunciator 3 für einen seriellen Austausch der Tastaturbelegung.

Das Tastaturlayout betrachte ich als nicht ganz gelungen. Meiner Meinung nach gehört links neben die „1" die „ESC"-Taste und nicht die Umschalttaste zwischen Programmier- und Normalmodus. Zu klein geraten sind die „TAB"-, „RETURN"-, „SHIFT"- und „DELETE"-Tasten. Geradezu eine Enttäuschung war jedoch, daß die „."- und „,"-Tasten vom Ziffernblock entfernt wurden, um die Apfeltasten unterzubringen. Gerade diese sind für gezielte numerische Eingaben dringend erforderlich, dringender noch als die Arithmetiktasten.

Ein Fauxpas ist auch, daß die „RESET"-Taste alleine zum Auslösen eines RESET ausreicht. Die Leute von Apple dachten sich schon etwas dabei, als sie die Tastenkombination „Control-RESET" dafür vorsahen. Wer beispielsweise im Wordstar statt auf die „DEL"-Taste aus Versehen auf die „RESET"-Taste kommt, wird sich wundern, was da alles „deleted" wurde. (Nämlich der gesamte Text!)
Vielleicht sollte man doch auf einige Funktionstasten verzichten und die Umschalttasten dorthin verlegen. Dann wäre Platz für größere „Shift-" und andere Sondertasten.

2. Interface und Einbau

Das Interface besteht aus einer Platine, auf der sich einige Steuerbausteine und ein großer Stecker befinden. Dieser Stecker wird in den Sockel des entfernten Keyboard-ROMs gesteckt. Das sind jedoch nicht alle Austauschaktionen, die vorgenommen werden müssen – fünf weitere ICs müssen herausgenommen werden, einer – der Keyboarddecoder – bleibt draußen, der Rest muß in einen Zwischensockel gesteckt werden, an den diverse Kabel angelötet sind.
Sind alle Umbauten beendet, schaltet man den Rechner ein und hofft, daß alles klappt. Wenn nicht, so bleibt zumindest dem Applehändler die Vorfreude auf eine

voll kostenpflichtige Reparatur wegen Fremdeingriffs.
Die fliegenden Kabel auf der Hauptplatine des Apple zeugen jedenfalls nicht von einer professionellen Konstruktion. Der Aufbau des Interface drückt den Gesamteindruck, den die Tastatur auf mich macht, erheblich.

3. Die Verwendung

Die Keycom 6 ist eine Tastatur für Tippfaule. Man kann jede Taste in 10 unabhängigen Ebenen mit einer Zeichenfolge belegen. Durch den Handshakeverkehr von Tastatur und Apple gehen nun wirklich keine Zeichen mehr verloren. Das Umschalten auf die verschiedenen Tastaturebenen geschieht über leicht einprägsame Tastenfolgen.
Einstellen kann man so ziemlich alles, was man wünscht. Selbst die Autorepeat-Rate kann über Software geregelt werden. Der Autorepeat kann für einzelne Tasten gesperrt werden etc. Eine eingehendere Beschreibung würde den Rahmen dieses Berichts sprengen.

Die Tasten sind angenehm zu betätigen, erlauben also ein langes, ermüdungsfreies Arbeiten mit der Tastatur. Nur die gegenüber der Appletastatur abweichende Belegung ist gewöhnungsbedürftig. Unverständlich bleibt mir allerdings, was das amerikanische „@"-Zeichen auf einer DIN-Tastatur verloren hat. Zur Erinnerung, im deutschen Zeichensatz ist es durch „§" ersetzt.
Der Ziffernblock ist eine echte Bereicherung der Tastatur. Er ermöglicht eine sicherere Eingabe von langen Zahlenkolonnen. Noch wertvoller wäre er, existierten noch die „."- und „,"-Tasten.

4. Software

Auf einer 5 1/4"-Diskette wird Software mitgeliefert, mit der man Tastaturbelegungen auf Diskette abspeichern und wieder zurückladen kann. Dies ist für CP/M und DOS 3.3 vorgesehen. Meiner Meinung nach sollten jedoch 10 Belegungsebenen pro Taste ausreichen.

5. Handbuch

Hoffentlich ist das Handbuch, das zum Testexemplar der Tastatur mitgeliefert wurde, eine Kopie des echten, sonst wäre es eine Zumutung. Der Text, von einem Matrixdrucker gedruckt – Schönschreibdrucker kosten ja so viel Geld – ist sehr ausführlich. Es läßt keine Wünsche offen, sowohl die Anwendung als auch der Aufbau werden detailliert geschildert. Nur ein Schaltplan fehlt noch, um die Sache abzurunden. Warum jedoch die Installationsanweisungen ans Ende des Handbuches verlegt wurden, bleibt der Spekulation überlassen. Am Anfang wären sie sicherlich besser aufgehoben, vor allem, weil man ohne genaueste Anweisungen schlicht und ergreifend verloren ist.

Fazit

Die Tastatur ist sicherlich ein hervorragendes Produkt. Einige Punkte bedürfen jedoch meiner Meinung nach der Verbesserung. So sollte das Interface, auch wenn das einen höheren Aufwand bedeutet, dort angeschlossen werden, wo es hingehört – an den Keyboard-Connector. Bedenklich erscheint mir das Herausreißen der verschiedenen ICs. Berücksichtigt man nun noch, daß die Firma Apple inzwischen einen Großteil dieser ICs direkt einlötet, ergibt sich nur eine Empfehlung:
Technisch Versierte können den Anschluß der Tastatur selbst vornehmen, allen anderen Anwendern bleibt nur der Weg zum Fachhändler, wenn er auch nur dem Erhalt der Garantie dient. Ein späterer Garantieausschluß wegen eines Fremdeingriffs kommt in jedem Fall teurer!
Ein weiterer Grund spricht für Änderungen: Es gibt inzwischen viele Apple IIe-kompatible Rechner, die teilweise andere ICs verwenden. An diese kann die Tastatur mit ihrem jetzigen Aufbau sicher nicht angeschlossen werden. Gerade Besitzer von kompatiblen Rechnern jedoch neigen zum Einsatz anderer Tastaturen. Viele Kompatible sind ja in IBM-ähnlichen Gehäusen untergebracht, eine externe Tastatur liegt da nahe.

TurtleGraphics-Library-Paket
von Dieter Geiß

Turtle-Utilities für Fenstertechnik und Apple-Maus in einfacher und doppelter Hires-Grafik für Pascal 1.2 auf Apple IIe/c mit Maus oder Joystick. 2 Disketten mit umfangreichem Manual, DM 98,–. Unter Pascal 1.1 mit 64K nur eingeschränkt lauffähig

Im einzelnen bietet das Paket folgende Möglichkeiten:
- volle Kompatibilität mit der alten „TurtleGraphics"
- alle zeitkritischen Funktionen in reinem Assembler programmiert
- Benutzung der zweiten Hires-Seite ($4000-$5FFF) möglich
- für Apple IIc und Apple IIe mit erweiterter 80-Zeichen-Karte Benutzung der doppelten Hires-Grafik mit 560 × 192 Punkten bzw. 16 neuen Farben möglich
- schnelle Prozeduren zum Zeichnen eines Punktes oder einer Linie
- Benutzung mehrerer Zeichensätze gleichzeitig
- Scrolling des Hires-Schirms oder eines Teils in vier Richtungen
- drei verschiedene Schriftarten: Fett-, Breit- und Proportionalschrift, beliebig mischbar (acht Möglichkeiten)
- spezielle schnelle Ausgabe von Text
- Cursor bei Eingabe frei programmierbar
- Ein-/Ausgabe von INTEGER-, CHAR-, STRING- und REAL-Werten im Grafikmodus
- Menüzeile wie beim Macintosh
- Pull-down-Menüs
- Laden und Speichern von Fenstern (Windows)
- Öffnen von Fenstern
- Aktivieren und Deaktivieren von Fenstern
- Verschieben und Vergrößern/Verkleinern von Fenstern
- Scrolling von Fensterinhalten in allen vier Richtungen
- Umfangreiche Demos als Quelltexte.

Hüthig Software Service · Postfach 10 28 69 · 6900 Heidelberg

DISK40
Disketten-Organisationsprogramm für Apple II+, IIe oder IIc

von Hermann Seibold und Dipl.-Ing. Udo Marin, 1986, Programmdiskette mit Anleitung, DM 48,–

DISK40 entstand aus der Analyse bestehender Kopierprogramme und vereint in sich eine Vielzahl von Möglichkeiten, die sich als nützlich erwiesen haben. Durch eine einfach zu bedienende Menüführung können DOS-3.3-Disketten umfangreich bearbeitet oder kopiert werden.
Zu den vielfältigen Möglichkeiten des Programms zählen u. a.:
- Tabellarische Ausgabe der Diskettenbelegung
- Ordnen des Catalogs
- „Undelete"n von versehentlich gelöschten Dateien
- Vergleichen von Disketten, Dateien oder der DOS-Spuren
- Kopieren von Disketten, Dateien oder DOS-Spuren
- Formatieren von Daten-Disketten
- Erweitern auf 40 Spuren bei bestehenden 35-Spur-Disketten
- Ändern des Boot-Programms
- File-Editor zum Editieren von Disketten-Dateien
- Komfortabler Sektor-Editor für Hex- und ASCII-Darstellung
- VTOC-Editor, z. B. zur Freigabe der DOS-Spuren

Schon nach wenigen Minuten können, dank der ausführlichen Beschreibung, Disketten nach eigenen Wünschen modifiziert oder Daten nach einem Disk-Crash wieder gerettet werden.

Hüthig Software Service · Postfach 102869 · Heidelberg 1

RGB-Farbinterfacekarte AI-80Z

Ein Erfahrungsbericht

von Dirk Katzschke

Allgemeines

Wer auf dem Apple II grafisch arbeitet, wird sich wohl schon einmal Gedanken über die Anschaffung eines Farbmonitors gemacht haben. Für die ersten „farbigen Versuche" mag der normale Farbfernseher ausreichen, wenn es dann aber darum geht, Text in 40 oder gar 80 Zeichen pro Zeile darzustellen, zeigen selbst Geräte neuerer Generation sehr schnell die Grenzen ihrer Leistungsfähigkeit. Auch die Darstellung auf einem entsprechenden Monitor mit Videoeingang ergibt aufgrund der nicht gerade hochwertigen Videoendstufe im Apple kaum bessere Ergebnisse. In dieser Situation bleibt dann eigentlich nur noch der Kauf eines RGB-Monitors und einer entsprechenden Interfacekarte übrig, da nur auf diese Art alle 16 Farben des Apple wirklich sauber darzustellen sind. Aber auch bei den RGB-Karten gibt es, wie bei vielen anderen Interfaces für den Apple, recht große Qualitätsunterschiede. Ein sehr gutes Ergebnis erzielte ich mit der AI-80Z der Firma NEC.

1. Einbau

Die AI-80Z (**Abb. 1**) wird mit einem Kabelset zum Anschluß im Rechner, einem Monitorkabel und einer 19seitigen Bedienungsanleitung geliefert. Mit Hilfe dieser Anleitung dürfte es selbst technisch sehr unerfahrenen Mitmenschen möglich sein, die Interfacekarte in Slot 7 einzubauen, sofern Sie in der Lage sind, rechts und links bzw. farbige Kabel zu unterscheiden. Die einzige Schwierigkeit beim Einbau in den Apple IIe besteht darin, daß ein IC aus seiner Fassung gezogen (z.B. mit einem kleinen Schraubenzieher oder einer Pinzette), in einen Zwischensockel eingesetzt und dann das Ganze wieder in die Fassung gesteckt werden muß. Besitzer eines Apple II+ können sich diese Mühe sparen, sollten aber bei der Bestellung angeben, daß Sie die Karte im II+ einsetzen wollen. In diesem Fall muß nämlich für die richtige Darstellung der Lores-Farben ein PROM auf der Karte ausgewechselt werden.

Die auf der Karte vorhandene Videobuchse wird bei der Verwendung einer Videx-Karte mit deren Videoausgang verbunden, Apple-IIe-80-Zeichen-Karten (mit und ohne einer beliebigen Speichererweiterung) werden automatisch erkannt. In beiden Fällen muß sich die betreffende Karte aber in Slot 3 oder dem Auxiliary Slot befinden.

2. Betrieb

Da es sich bei der AI-80Z um eine reine Hardware-Karte handelt, werden an den Systemausbau des Apple II+ oder IIe keinerlei Anforderungen gestellt. Am Monitorausgang der Karte wird allerdings nur ein TTL-RGB-Signal zur Verfügung gestellt, weshalb die Verwendung eines 4-Bit-TTL-RGB-Monitors erforderlich ist.

Die Grundeinstellung der Karte erfolgt mit 8 DIP-Schaltern, jedoch können alle wichtigen Funktionen über die 16 Device-Select-Adressen von Slot 7 ($C0F0-$C0FF = 49392-49407) per Software gesteuert werden (**Tabelle 1**). Der Standardwert der Farbe – nach dem Einschalten des Rechners oder einem Reset – ist allerdings immer weiße Schrift auf schwarzem Hintergrund.

2.1. Textdarstellung

Bei der Darstellung von 40 oder 80 Zeichen kann aus den 16 Lores-Farben für Text und Hintergrund gewählt werden, wodurch sich prinzipiell 256 verschiedene Möglichkeiten ergeben. In der Praxis sind allerdings einige Kombinationen recht *„nutzlos"* (z.B. Blau auf Blau, Hellgrün auf Dunkelgrün usw.). Wenn man all jene Möglichkeiten aus der Tabelle der Farbwahl streicht, bleiben aber immer noch über 150 verwertbare Kombinationen übrig. Damit könnte sich der Apple II durchaus als preiswertes Textverarbeitungssystem in kleineren Betrieben oder Schulen durchsetzen, denn die von vielen bemängelte Anpassungsfähigkeit der Bildschirmdarstellung an die verschiedensten Arbeitsplätze stellt bei einer solchen Auswahl

Tabelle 1

Poke 49392,nn ($C0F0)	: das Farbregister wird mit der Farbkombination nn gesetzt
Poke 49400,nn ($C0F8)	: Double-Density HGR-Modus ein
Poke 49401,nn ($C0F9)	: Normal HGR-Modus
Poke 49402,nn ($C0FA)	: Farbdarstellung ein (Standard)
Poke 49403,nn ($C0FB)	: Farbdarstellung aus <1>
Poke 49404,nn ($C0FC)	: 40- oder 80-Zeichen-Darstellung möglich <2>
Poke 49405,nn ($C0FD)	: nur 40-Zeichen-Darstellung möglich <2>
Poke 49406,nn ($C0FE)	: Farbregister anwählbar (Standard)
Poke 49407,nn ($C0FF)	: Farbregister gesperrt

<1> Text, Lores- und Hires-Grafik werden monochrom in der mit nn ausgewählten Farbkombination dargestellt.
<2> nur Apple II+. Im IIe ist dieser Softswitch wirkungslos. DIP Schalter 6 sollte deshalb auf OFF stehen.

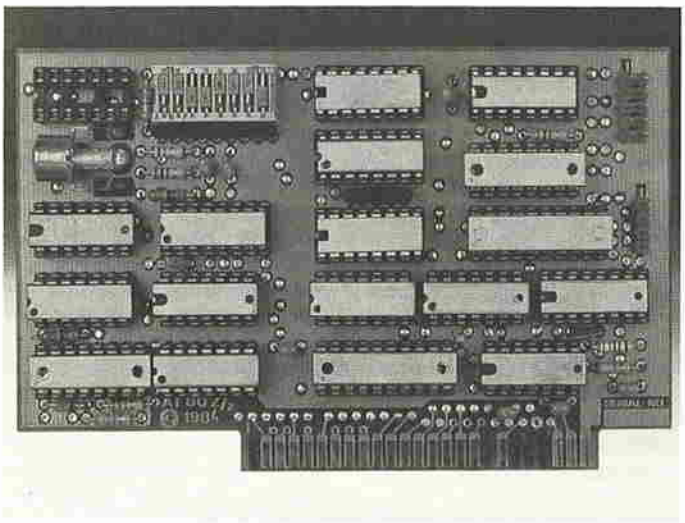


Abb. 1: AI-80Z

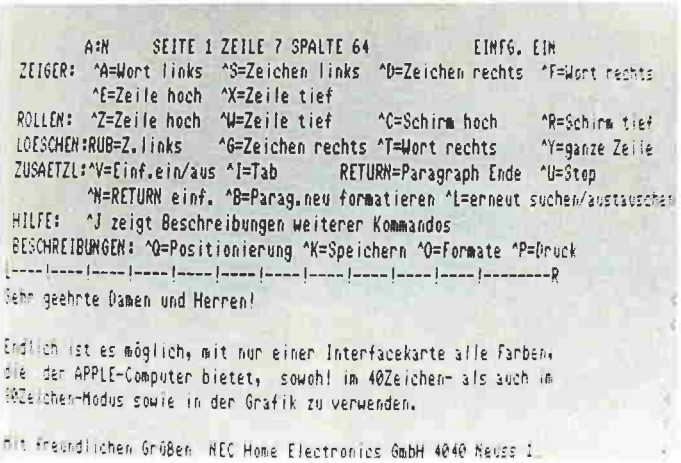


Abb. 2: Textdarstellung

Abb. 3: Farbskala

wohl kein so großes Problem mehr dar. Wenn Sie das Photo der Textdarstellung (**Abb. 2**) einmal genau betrachten, werden Sie feststellen, daß endlich ein altes Apple-Problem gelöst ist; die Überbetonung der horizontalen Linien ist verschwunden. Als sehr angenehm bei Textverarbeitung empfand ich, daß die Hintergrundfarbe auf dem gesamten Schirm und nicht nur im Textfenster dargestellt wird. Bei der Textdarstellung werden, sowohl bei 40- als auch bei 80-Zeichen-Darstellung, die einzelnen Buchstaben ohne jeden Farbrand dargestellt. Dies ist leider bei den meisten RGB-Karten keinesfalls so selbstverständlich, wie man annehmen sollte. Ich kann Ihnen, sofern Sie im Besitz eines Farbmonitors sind, nur raten, selbst einmal eine kleine Lupe zu nehmen und die Farbreinheit der Darstellung bei allen darstellbaren Farben zu prüfen – Sie werden sich wundern...

2.2. Grafik

Wenn auch die Möglichkeiten der Textdarstellung recht vielseitig und praktisch sind – nicht zuletzt deshalb, weil man keinen zweiten monochromen Monitor zur Textdarstellung betreiben muß und Platz auf dem Schreibtisch spart (bei meinem Schreibtisch ein sehr wichtiger Punkt) – so wird die AI-80Z wohl hauptsächlich wegen ihrer Grafikmöglichkeiten angeschafft werden. Allerdings sollte man sich darüber klar sein, daß man aufgrund der etwas obskuren Art der Grafikdarstellung des Apple keine Wunder erwarten darf. Und an der für heutige Verhältnisse etwas mageren Auflösung kann auch eine noch so ausgefeilte RGB-Karte nichts ändern. Was aber mit Hilfe der AI-80Z aus der Grafik herauszuholen ist, braucht sich nicht zu verstecken.

Bei der Darstellung der Lores-Farben (**Abb. 3 und 4**) fielen mir zwei Dinge besonders auf: schmale Farbsäume bei einigen Farben und die Darstellung der Farbe Braun. Die Farbsäume stellten sich nach Rückfrage bei der Firma NEC in Neuss nicht als Fehler der Karte oder des Monitors, sondern als durchaus beabsichtigt heraus. „...schließlich ist dies von der Firma Apple so vorgesehen, und es kann ja durchaus sein, daß jemand diesen Effekt ausnutzt...warum sollten wir also etwas daran ändern...", war die Antwort eines dortigen Technikers. Die Farbe Braun löste nicht nur bei mir, sondern auch bei dem Techniker unseres Institutes Verwunderung aus. Bis jetzt war mir nicht bekannt, daß es mit einem TTL-RGB-Monitor möglich ist, ein „vernünftiges" Braun darzustellen. Aber wie man sieht, funktioniert es. Doch fragen Sie mich bitte nicht, warum und wie. An dieser Stelle sei mir die Bemerkung erlaubt, daß man mir gegenüber bei der Firma NEC durchaus großzügig in punkto technischer Informationen zur RGB-Karte war und ist (leider ist dies heutzutage nicht mehr der Normalfall...), sich allerdings nicht in die „Trickkiste" schauen ließ.

Double Lores ist auf einem Apple IIe mit erweiterter 80-Zeichenkarte ohne weiteres möglich, obwohl die AI-80Z zu einem Zeitpunkt entwickelt wurde, als an den IIe noch nicht zu denken war.

Bei der normalen Hires-Darstellung (**Abb. 5**) bemerkt man eine leichte Schraffur aller sechs Hires-Farben, da diese auf dem Apple – wie wohl hinlänglich bekannt ist – nicht mit jedem einzelnen Punkt dargestellt werden können. Da dies bei Spiel-Grafiken etwas stören kann, gibt es den sogenanten Double-Density-HGR-Modus. Allerdings ist dabei sowohl die Darstellung von Text auf der Hires-Seite als auch die Farbqualität nicht mehr ganz so sauber, jedoch erhält man wirklich ausgefüllte Farbflächen, womit sich dieser Modus wohl am besten für Adventure-Spiele mit größeren Farbflächen eignen dürfte. Ein Bekannter nannte diesen Modus mit der Bemerkung – „...das sieht jetzt genauso aus wie bei meiner Taxan-RGB-Karte..." einfach den Taxan-Modus. (Besitzer jener Karte mögen ihm diesen Vergleich verzeihen.)

Abb. 5: Hires-Beispiel

Abb. 4: Lores-Beispiel

Wenn man z.B. mathematische Funktionen auf der Hires-Seite darstellen und diese dann auch noch beschriften will, ergibt sich bei einem Farbmonitor das Problem, daß eine weiße Linie keinesfalls auch überall weiß dargestellt wird. Statt dessen erhält man ein recht buntes Ergebnis. Dieses Problem wurde bei der AI-80Z sehr elegant gelöst. Nach dem Ansprechen der Speicherstelle $C0FB (49403) wird die Lores- bzw. Hires-Grafik monochrom dargestellt, und man sieht anstatt der Farben die von monochromen Monitoren her gewohnten Schraffuren. Dabei kann man für die Darstellung der Grafik-Seiten wiederum aus

den 16 Lores-Farben für Hintergrund und Grafik wählen. Dadurch ergeben sich auch hier 256 verschiedene Möglichkeiten. Die Textdarstellung des Programms HRCG (DOS-3.3-Toolkit-Diskette) ist in diesem Modus praktisch nicht mehr von dem der 40-Zeichen-Textseite zu unterscheiden.

Eine kleine Einschränkung gibt es bei der Darstellung von Double-Hires-Grafiken. Im normalen HGR-Modus lassen sich z.Zt. nur 8 der 16 Farben darstellen. Will man alle 16 Farben erhalten, bleibt aber noch der Double-Density-HGR-Modus. Die Turtle-Grafik von Herrn Geiß zeigte allerdings auch in diesem Modus noch beachtliche Ergebnisse. Mit dem Programm SUPERPLOT konnte sogar eine optisch bessere Darstellung als auf einem monochromen Monitor erreicht werden, da auch im monochromen HGR-Modus die Überbetonung der horizontalen Linien entfällt und zusätzlich eine für das menschliche Auge angenehmere Farbkombination als Grün/Schwarz gewählt werden konnte.

3. Betriebssysteme

Mit allen apple-eigenen Betriebssystemen läuft die AI-80Z völlig problemlos. Lediglich CP/M (in diesem Fall Version 2.2 auf der AP22 von IBS) und das UCSD p-System zur AP20 des gleichen Herstellers machten ein paar Schwierigkeiten.

Beim Booten von CP/M schaltete die AI-80Z auf grün-schwarze Darstellung um. Dieser Effekt konnte jedoch durch Sperren des Farbregisters (siehe Tabelle 1) und Neubooten mit PR#6 behoben werden. Auf einem enhanced Apple IIe ist, bei Verwendung der 160-Track-CP/M-Version und in Verbindung mit einem Ehring-FDC4-Controller, das Sperren des Farbregisters sogar unbedingt erforderlich. Ansonsten erhält man nämlich nur die lapidare Mitteilung: „Bios Positionierfehler auf Drive A:".

Die Schwierigkeiten mit dem p-System der 68000-Karte waren allerdings etwas verzwickter. Wurde dieses Betriebssystem gestartet, verschwand nach dem Aktivieren der 80-Zeichen-Karte (IIe) jeglicher Text vom Bildschirm, das System arbeitete jedoch einwandfrei. Dieser Zustand ließ sich nur durch Control-Reset wieder aufheben, wodurch neu gebootet wurde und das Spielchen von vorne begann. Dieses Verhalten war nicht nur für mich, sondern auch für die Firmen NEC und FOCUS Computer (von ihr habe ich das p-System erhalten) nicht zu erklären. Mehr oder minder zufällig gelang es mir, den selben Effekt mit folgender BASIC-Zeile zu erzeugen:

PR#3 : POKE 49328,0

Weiteres Probieren zeigte, daß jedes Ansprechen einer Device-Select-Adresse von Slot 3 ($C0B0-$C0BF) bei aktiver 80-Zeichen-Firmware diese „Verdunkelung" auslöste. Nach dem Abschalten der 80-Zeichen-Karte war das Bild jeweils wieder da. Als eigentliche Fehlerquelle erwies sich schließlich der DIP-Schalter Nr. 6 auf der AI-80Z. Laut Bedienungsanleitung sollte dieser DIP-Schalter im Normalfall auf ON stehen. Diese Einstellung wird von der IIe-80-Zeichen-Karte jedoch ignoriert. Damit die Device-Select-Adressen von Slot 3 von der AI-80Z nicht mehr geprüft werden, muß dieser DIP-Schalter im Apple IIe auf OFF stehen. Danach funktionierte auch die 68000-Karte einwandfrei.

Abschließend noch ein kleiner Tip für diejenigen von Ihnen, die sich mit einem Lötkolben an Ihren Apple herantrauen. Da die AI-80Z nur die Device-Select-Adressen von Slot 7 benutzt, ist es durchaus möglich, daß sie sich den Slot z.B. mit einer Z80-A-Karte – die diese 16 Adressen nicht benutzt – „teilt". Dazu muß allerdings ein zusätzlicher Slot zu Slot 7 parallel geschaltet werden. Dabei ist wegen des sehr kritischen Zeitverhaltens der Karte auf möglichst kurze Verbindungen zu achten. Besitzer eines IIe-kompatiblen Rechners könnten auch den eigentlichen Slot 3 für die Karte verwenden, müßten dabei aber auf zwei Dinge achten: erstens fehlen im Slot 3 die Signale Video 3.58 MHz (Pin 35) und Sync (Pin 19), müssen also von Slot 7 „beschafft" werden, und zweitens verschieben sich natürlich die Speicherstellen, mit denen die Funktionen der AI-80Z gesteuert werden können. Da bei einem solchen Umbau aber die Garantie der Karte erlischt, sollten Sie erst nach Ablauf der Garantiefrist daran denken.

Fazit

Wer auf der Suche nach einer vielseitigen RGB-Karte für seinen Apple ist, sollte ruhig einmal einen Blick auf die AI-80Z werfen. Dabei wird er sehr bald feststellen, daß es für „richtige" Grafik nicht immer gleich einer jener Rechner mit den berühmten 3 Buchstaben (oder eines der vielen Plagiate) sein muß. Schließlich kann ein IBM PC in der Standard-Grafik eher weniger als der Apple. Ich frage mich nur, warum die Firma Apple nicht schon längst selber auf die Idee zu einer solchen RGB-Karte gekommen ist. Doch leider ist man es in letzter Zeit ja fast schon gewohnt – gute Ideen haben zuerst ... andere.

INPUT 2.0

Ein Bildschirm-Maskengenerator für DOS 3.3 und ProDOS
von U. Stiehl

1984, Diskette und Manual, DM 98,–
ISBN 3-7785-1021-5

Der für den Apple II bestimmte Maskengenerator „Input 2.0" basiert auf den früheren Programmen „Input 1.0" und „Input 80 1.0" (von denen noch Restbestände lieferbar sind) und ist sowohl unter DOS 3.3 wie auch unter dem neuen ProDOS lauffähig. Der Maskengenerator setzt einen Apple II Plus mit Language Card oder einen Apple IIe voraus. Im 40 Z/Z-Modus funktioniert er auf beiden Gerätetypen, im 80 Z/Z-Modus dagegen nur auf dem Apple IIe mit 80-Zeichen-Karte. (Die alte Videx-Karte für den Apple II wird nicht unterstützt!)

„Input 2.0" liegt wahlweise in der Bank 1 oder Bank 2 der Language Card und wird durch einen kurzen Driver in den unteren 48K aufgerufen. „Input 2.0" läßt sich problemlos in nicht-compilierte und compilierte Applesoft- sowie in Assemblerprogramme einbinden. Die Übergabe der Feldinhalte an das Anwenderprogramm erfolgt durch ein einfaches Verfahren, das auch bei Compilern funktioniert.

Für jedes Feld der Bildschirmmaske lassen sich u. a. definieren: Feldlänge (bis zu 255 Zeichen) – Vtab – Htab – Datentyp (insgesamt 8 Typen) – Scrollflag (starre oder dynamische Maske) – Ctrlflag – Füllflag – Löschflag – Bildschirmflag (40- oder 80 Zeichendarstellung). Innerhalb eines Eingabefeldes besteht jeder denkbare Redigierkomfort (Insert, Delete, Rubout, Restore usw.).

Bei der neuen Version des Maskengenerators können jetzt auch Ctrl-Zeichen beim Datentyp String eingegeben werden. Ferner sind – das gilt nur für IIe – die Apfeltasten als schnelle Cursortasten definiert.

Gerätevoraussetzung: Apple IIe oder IIc; ferner Apple II+ im 40-Zeichenmodus

Hüthig Software Service,
Postfach 10 28 69,
D-6900 Heidelberg

In Kürze zusätzlich als Quellcode für Kyan-Pascal lieferbar, Preis DM 28,–.

Star Gemini-15

getestet von Jürgen Geiß

Beim SG-15 (die breite Version des SG-10) handelt es sich um einen Dot-Matrix-Drucker, die wohl am weitesten verbreitete Technik für Drucker der unteren Preisklasse. Dadurch können mit ihm auch DIN-A4-Querformate verarbeitet werden.

Die Hardware

Das Zusammenbauen des frisch ausgepackten Druckers ist kein Problem, lediglich das Farbband, ein gewöhnliches Schreibmaschinenfarbband mit 2 Spulen, läßt sich, ohne schwarze Finger zu bekommen, nicht so einfach einfädeln. Man muß sich durch etliche Windungen, Ösen und Bandführungsbolzen schlängeln. Die Lösung bei Epson ist beispielsweise besser: eine einfach einzulegende Kassette, bei der das Farbband nicht einmal berührt werden muß.
Besitzt man einen Computer mit Centronics-(Parallel)-Schnittstelle, so kann man den SG-15 sofort anschließen und drucken. Als Option wird auch eine RS-232C-(V24)-Schnittstelle angeboten.
Der Drucker hat eine Geschwindigkeit von 120 Zeichen/s bei Pica-Normalschrift und druckt bidirektional mit Druckwegoptimierung. Das ist heutzutage nichts Aufregendes, denn andere bringen es bis auf 200 Z/s oder mehr. Die Geschwindigkeit ist aber ausreichend, wenn man bedenkt, daß der SG-15 ein besonderes Bonbon bereithält. Er besitzt intern „satte" 16K Druckerpuffer, so daß schon längere Listings gedruckt werden können, während am Computer weitergearbeitet werden kann. Die meisten anderen Hersteller bieten nur 80 Zeichen oder 2K Puffer in der Standardausführung. Für Anwenderprogramme, bei denen des öfteren große Datenmengen zu drucken sind, ist er also bestens vorbereitet.
Der SG-15 besitzt sowohl eine Friktionswalze für Einzelblattpapier als auch einen Traktor für Endlospapier. Der Traktor läßt sich leicht mit einem Griff abnehmen und wieder aufsetzen. Leider sitzt er obenauf wie etwa bei den älteren Epson-Druckern der MX-Serie. Dadurch wird immer eine Seite Papier verschenkt, wenn man diese nach einem Druckvorgang an der Perforation abreißen möchte. Eine Verlegung des Traktors unterhalb der Walze hätte diesen Nachteil behoben und zudem noch den Vorteil gehabt, einen Zeilenrückschub einbauen zu können. Das Einfädeln von Endlospapier stellte sich auch nicht gerade als problemfrei heraus. Wenn man das Papier unter der Walze durch zum oben liegenden Traktor schieben will, braucht man schon etwas Geduld.
Der Drucker besitzt zwei DIP-Schalterblöcke, einen mit 8, den anderen mit 4 Schaltern. Sie sind von außen sehr gut an der linken Druckerseite zugänglich. Damit lassen sich Grundeinstellungen wie Zeichensatz, Schriftart usw. einstellen. Ein Schalter schaltet auf IBM-Kompatibilität um, so daß sich ein IBM-PC problemlos anschließen läßt.

Die Software

Am besten fängt man mit dem eingebauten Selbsttest an. Dieser wird durch Einschalten der Drukkers und gleichzeitiges Betätigen der LF-Taste aktiviert. Der SG-15 druckt dann in 8 Zeilen seine gesamten Zeichensätze aus. Hierbei sind die Druckgeräusche etwas lauter, als man dies von Epson oder dem Imagewriter her kennt. Diese bestehen aus dem ASCII-Satz und weiteren 7 internationalen Zeichensätzen wie Frankreich, Deutschland, England usw. und einigen Grafikzeichen. Schaltet man auf IBM-Modus um, so wird mit dem speziellen IBM-Satz gedruckt. Außerdem besitzt er noch entsprechende Kursiv- und Schönschreibzeichensätze (Near Letter Quality). Diese Schönschrift kann sich durchaus sehen lassen, denn man kann die einzelnen Drucknadeln tatsächlich nicht mehr erkennen. Dies erkauft man sich natürlich mit einer geringeren Druckgeschwindigkeit.
Wem diese Zeichensätze nicht genügen, dem kann geholfen werden. Wie bei jedem besseren Drucker können auch hier selbstdefinierte Zeichensätze erstellt werden. Bis zu 240 eigene Zeichen passen dann in den Download-Zeichenspeicher des Druckers Die Schriftarten sind mannigfaltig und reichen von breit, fett über Doppeldruck, proportional, kursiv bis zur Schönschrift. Auch ein Unterstreichungsmodus wird geboten. Damit sind die Möglichkeiten des Druckers natürlich noch lange nicht erschöpft. Wem der Textmodus immer noch nicht das Richtige bietet, für den gibt es selbstverständlich mehrere Grafikmodi. Im Grafikbetrieb lassen sich 8 Bits Daten senden, die dann als vertikales Punktraster interpretiert werden. Insgesamt gibt es 7 verschiede Modi: 480 Punkte/Zeile, 960, 960 mit doppelter Geschwindigkeit, 1920, 640, 576 und 720 Punkte/Zeile. Wer einen Epson-Drucker besitzt, dem kommen diese Zahlen sicher sehr bekannt vor.
Des weiteren bietet der SG-15 noch einen Hex-Dump-Modus, bei dem alle ankommenden Zeichen an den Drucker nicht als ASCII-Zeichen interpretiert, sondern als hexadezimale Ziffern ausgegeben werden. So wird bei einem „A" z.B. ein „41" ausgegeben. Dies entspricht der Dezimalzahl 65, also genau einem „A" in der ASCII-Tabelle. Dieser Modus hilft sehr bei der Fehlersuche von selbstentwickelten Programmen.
Zum Lieferumfang gehört natürlich ein Handbuch, das in deutscher Sprache verfaßt wurde. Dort werden alle Befehle des Druckers mit Beispielen erläutert. Allerdings ist das Handbuch etwas unübersichtlich, denn im Star-Modus gelten meist andere Steuerzeichen als im IBM-Modus, so daß man für die gleiche Funktion meist 2 verschiedene Steuerzeichen im Kopf haben muß.

Der Preis des SG-15 betrug Mitte August 1985 DM 1650,-. Dies ist recht günstig für einen Drucker mit einer Wagenbreite von 15,5 Zoll und einem Puffer von 16K. Vor dem Kauf sollte man aber Vor- und Nachteile sorgfältig abwägen. Wenn Sie technische Fragen haben, z.B. Grafikdump-Probleme usw., so wenden Sie sich bitte direkt an die Firma Star Europa GmbH oder deren Distributoren und nicht an den Peeker, denn die Peeker-Redaktion sendet nach dem Abschluß von Testberichten die Hard- und/oder Software grundsätzlich wieder an die jeweilige Firma zurück.

Redakteur

Für unsere Zeitschrift Peeker suchen wir einen Redakteur mit soliden Assembler-Programmierkenntnissen.
Es erwartet Sie eine abwechslungsvolle Tätigkeit in einem großen Verlagshaus.

Ihre Bewerbung mit den üblichen Unterlagen richten Sie bitte an:

Verlagsgruppe Dr. Alfred Hüthig
Personalabteilung
Postfach 10 28 69
6900 Heidelberg

Für engagierte Anwender:

Mit mc lernen Sie Computer von Grund auf verstehen. Ausführliche Funktionsbeschreibungen von Rechner-Hardware und gut kommentierte Programm-Listings bieten Ihnen den richtigen Einstieg ins ernsthafte Computern.

Durch Programme in mc werden Sie manches Problem überhaupt nicht mehr als Problem betrachten.

Nach mc-Bauanleitungen löten Sie vom einfachen Interface bis zum kompletten System, was an Hardware nur schwer zu kaufen ist.

In mc-Fachaufsätzen geht's um neue Entwicklungen, um professionelle Hardware und Peripherie.

Natürlich testet mc Geräte und Programme. Die Ergebnisse werden aus der Sicht des professionellen Anwenders interpretiert.

Aktuelles aus der Branche zu Unternehmen, Produkten, Kongressen, Tagungen und Messen finden Sie jeden Monat in mc.

mc erscheint monatlich, das Einzelheft kostet DM 6,50 und ist an jeder größeren Zeitschriftenverkaufsstelle zu haben. Der Jahresabonnementspreis beträgt DM 66,–, im Ausland DM 72,– für 12 Hefte pro Jahr.

Garantie: Wenn Sie von unserem Kennenlern-Angebot Gebrauch machen, können Sie innerhalb von 10 Tagen nach Erhalt des kostenlosen Heftes ein Abonnement widerrufen. Zur Wahrung der Frist genügt rechtzeitiges Absenden an den Franzis-Verlag, Postfach 37 02 80, 8000 München 37.

mc
Die Mikrocomputer-Zeitschrift

Kennenlern-Angebot

Ich möchte die neueste Ausgabe der mc kostenlos zum Probelesen. Informiere ich Sie nach Erhalt des kostenlosen Heftes nicht anders, abonniere ich die mc zum vorteilhaften Abonnementspreis.

Garantie: Die Abonnementsbestellung kann ich innerhalb von 10 Tagen nach Erhalt des kostenlosen Heftes beim Franzis-Verlag, Postfach 37 02 80, 8000 München 37, widerrufen. Zur Wahrung der Frist genügt rechtzeitiges Absenden.

Datum

Unterschrift

Wenn Sie nichts mehr von mir hören, möchte ich ab _____ Ihre Zeitschrift **mc** 12 Hefte pro Jahr, Jahresabonnementspreis DM 66,–, im Ausland DM 72,–, abonnieren. Senden Sie die Hefte an folgende Anschrift:

Name/Vorname

Beruf

Straße

PLZ/Ort

Datum Unterschrift A 4 PE

In den genannten Abonnementspreisen sind sämtliche Nebenkosten, einschließlich Porto, enthalten. Die Kündigung ist jederzeit zum Ende des bezahlten Zeitraumes möglich. Die Abonnementsgebühr ist nach Erhalt der Rechnung fällig.

Franzis' Franzis-Verlag, Postfach 37 02 80, 8000 München 37

Warum wollen Sie ein teures Textverarbeitungsprogramm kaufen, wenn es ein billiges und besseres gibt?

Fast-Writer

von Harald Grumser
Programmdiskette und Handbuch
Gerätevoraussetzung: Apple IIe oder IIc (nicht II+)

DOS-3.3-Version. Auslieferung ab 1.6.86
Normalpreis DM 128,– (ISBN 3-7785-1419-9)
Sonderpreis für Peeker-Abonnenten DM 98,–

ProDOS-Version. Auslieferung ab 1.9.86
Normalpreis DM 128,– (ISBN 3-7785-1421-0)
Sonderpreis für Peeker-Abonnenten DM 98,–
Kombinationspreis für Bezieher der DOS-3.3-Version DM 28,–

Der Fast-Writer von Harald Grumser ist in zahlreichen Funktionen wie Scrollen, Suchen und Ersetzen mit Abstand das schnellste und damit angenehmste Textverarbeitungsprogramm für den Apple IIe oder IIc.

Flexibilität

Viele Textverarbeitungsprogramme sind geschützt und laufen deshalb nur in Verbindung mit normalen Disk-II-Laufwerken. Nicht so der Fast-Writer!

– Der Fast-Writer modifiziert weder DOS 3.3 (oder Diversi-DOS) noch ProDOS und kann deshalb mit BRUN FAST.WRITER gestartet werden. Unter Diversi-DOS ist der Fast-Writer dann in 3 Sekunden im Speicher. Vergleichen Sie einmal, wie lange es dauert, bis andere Textprogramme im Speicher sind!

– Da der DOS-3.3-Fast-Writer in den oberen 16K (= Language Card) liegt, kann man ihn vorübergehend verlassen und mit einem einfachen Befehl wieder starten. Mit anderen Worten: Der Fast-Writer ist permanent verfügbar, auch wenn Sie zwischendurch beispielsweise mit FID Dateien kopiert haben.

– Der Fast-Writer läuft mit allen externen Datenspeichern, die für DOS 3.3 oder ProDOS gedacht sind, z.B. mit dem Erphi-160-Spur-Subsystem, mit der Megaboard-MDB-Festplatte, mit RAM-Karten usw. Spezielle Anpassungen sind nicht erforderlich. Suchen Sie einmal ein Textprogramm, das mit diesen Datenspeichern auf Anhieb funktioniert!

– Der Fast-Writer kann mühelos über ein Menü für Ihre speziellen Aufgaben konfiguriert werden. Sie können z.B. per Knopfdruck die Zeilenbreite (normal 80 Zeichen) am Bildschirm einstellen, wobei ab einer Breite von weniger als 41 Zeichen automatisch auf die größere Bildschirmschrift umgestellt wird. Ferner können Sie die Größe des Arbeitsspeichers (insgesamt ca. 35 500 Zeichen) beliebig in Textspeicher und Hilfspuffer (für Löschen und Blockverschieben) aufteilen. Wenn Sie z.B. große Textblöcke im Speicher zu verschieben haben, so können Sie einen entsprechend großen Hilfspuffer von z.B. 10 000 Zeichen einrichten. Damit entfällt das zeitraubende Zwischenspeichern und Einlesen von Diskette.

Befehlsvorrat

Der Fast-Writer verfügt über eine große Zahl von Befehlen, von denen Sie in der Praxis jedoch nur wenige benötigen. Fünf Befehlsübersichten sind durch eingebaute Hilfsübersichten immer abrufbar, so daß Sie schon nach einer mehrstündigen Einarbeitung auf das Handbuch verzichten können. Eine Auswahl der wichtigsten Befehle:

– Freie Cursorbewegung in allen vier Richtungen mit eingebauter Schnell-Scroll-Routine.

– Diverse, per Knopfdruck ein- und ausschaltbare Optionen, z.B. Wortumbruch/kein Wortumbruch, Return sichtbar/unsichtbar, Kopfzeile (Statuszeile) mit Speicherbelegung, Cursorposition usw. eingeblendet/ausgeblendet, Bildschirm geteilt/ungeteilt, Tabulatorleiste sichtbar/unsichtbar, überschreibmodus (statt normalen Einfügmodus) ein/aus usw.

– Eingabe von Kontrollbuchstaben (einschließlich Ctrl-V!) möglich. Automatische Konvertierung in Groß- oder Kleinschreibung (unter Berücksichtigung der Umlaute und ß!)

– Extrem schnelles Suchen und Ersetzen von Zeichenketten (vorwärts und rückwärts).

– Makros frei definierbar und per Knopfdruck abrufbar. Makros können nicht nur stereotype Wortfolgen sein (z.B. „Sehr geehrte Herren"), sondern auch alle Befehlsfolgen, die man beim Fast-Writer sonst über die Tastatur eingeben würde. So läßt sich beispielsweise ein Text automatisch von Laufwerk 1 laden und auf Laufwerk 2 speichern.

– DOS-Kommandos wie Catalog, Delete, Rename usw. immer verfügbar (bei DOS 3.3 zusätzlich Init, bei ProDOS zusätzlich Online und Datum)

– Ausdruck auf Matrixdrucker (normal endlos), Schreibmaschine (normal mit Einzelblatt) und zu Kontrollzwekken auf Bildschirm; links- und rechtsbündig, zentriert und Blocksatz; einstellbarer linker, rechter und oberer Rand (im Text änderbar), bei Bedarf mit Kopfzeile und Paginierung usw. Der Ausdruck kann über eigene Druckertreiber umgelenkt werden, um z.B. Probleme mit Typenrädern, Steuerzeichen usw. zu beheben.

– Makros, Druckparameter, Druckertreiber und Tabulatoren können auf Diskette gespeichert werden.

Hüthig Software Service · Postfach 10 28 69 · 6900 Heidelberg 1